NetModule Router NB3700

User Manual for Software Version 3.7

Manual Version 1.3

NetModule AG, Switzerland

October 29, 2014

Contents

1 Welcome to NetModule **3**

2 Conformity **4**
2.1 Safety Instructions . . . 4
2.2 Declaration of Conformity . . . 5
2.3 Waste Disposal . . . 5
2.4 National Restrictions . . . 5
2.4.1 France . . . 5
2.4.2 Italy . . . 6
2.4.3 Latvia . . . 6
2.4.4 Luxembourg . . . 6
2.4.5 Norway . . . 6
2.4.6 Russian Federation . . . 6
2.4.7 Turkey . . . 7

3 Specifications **8**
3.1 Features . . . 8
3.2 Operating Elements . . . 9
3.3 Interfaces . . . 11
3.3.1 Overview . . . 11
3.3.2 USB 2.0 Host Port . . . 12
3.3.3 M12 Ethernet Connectors . . . 12
3.3.4 Power . . . 12
3.3.5 Digital Inputs and Outputs . . . 13
3.3.6 RS-232 Port . . . 14

4 Installation **16**
4.1 Environmental Conditions . . . 16
4.2 Installation of the Router . . . 16
4.3 Installation of the SIM Card . . . 16
4.4 Installation of the WLAN Antennas . . . 17
4.5 Installation of the Local Area Network . . . 17
4.6 Installation of the Power Supply . . . 17

5 Configuration **18**
5.1 First Steps . . . 18
5.1.1 Initial Access . . . 19
5.1.2 Recovery . . . 19
5.2 HOME . . . 20
5.3 INTERFACES . . . 22
5.3.1 WAN . . . 22
5.3.2 Ethernet . . . 29
5.3.3 Mobile . . . 35
5.3.4 WLAN . . . 41
5.3.5 USB . . . 49
5.3.6 Serial Port . . . 52
5.3.7 Digital I/O . . . 54
5.3.8 GNSS . . . 55
5.4 ROUTING . . . 57
5.4.1 Static Routes . . . 57
5.4.2 Extended Routing . . . 59
5.4.3 Multipath Routes . . . 61
5.4.4 Mobile IP . . . 62
5.4.5 Quality Of Service . . . 66
5.5 FIREWALL . . . 68
5.5.1 Administration . . . 68
5.5.2 Adress Groups . . . 68
5.5.3 Rules . . . 68
5.5.4 NAPT . . . 70
5.6 VPN . . . 74
5.6.1 OpenVPN . . . 74
5.6.2 IPsec . . . 79
5.6.3 PPTP . . . 85
5.6.4 GRE . . . 88
5.6.5 Dial-In . . . 89
5.7 SERVICES . . . 91
5.7.1 SDK . . . 91
5.7.2 DHCP Server . . . 101
5.7.3 DNS Server . . . 104
5.7.4 NTP Server . . . 105
5.7.5 DynDNS . . . 106
5.7.6 E-Mail . . . 108
5.7.7 Events . . . 110
5.7.8 SMS . . . 111
5.7.9 SSH/Telnet Server . . . 114
5.7.10 SNMP Agent . . . 116

5.7.11 SNMP Configuration . . . 118
5.7.12 SNMP Authentication . . . 119
5.7.13 Web Server . . . 120
5.7.14 Redundancy . . . 121
5.7.15 Voice Gateway . . . 123
5.8 SYSTEM . . . 125
5.8.1 System . . . 125
5.8.2 Authentication . . . 129
5.8.3 Software Update . . . 132
5.8.4 Configuration . . . 134
5.8.5 Troubleshooting . . . 138
5.8.6 Keys and Certificates . . . 141
5.8.7 Licensing . . . 144
5.8.8 Legal Notice . . . 145
5.9 LOGOUT . . . 146

6 Command Line Interface 147
6.1 General Usage . . . 147
6.2 Print Help . . . 148
6.3 Getting Config Parameters . . . 149
6.4 Setting Config Parameters . . . 149
6.5 Getting Status Information . . . 150
6.6 Scanning Networks . . . 151
6.7 Sending E-Mail or SMS . . . 151
6.8 Updating System Facilities . . . 151
6.9 Restarting Services . . . 152
6.10 Debug System . . . 153
6.11 Resetting System . . . 153
6.12 Rebooting System . . . 154
6.13 Running Shell Commands . . . 154
6.14 Working with History . . . 154
6.15 CLI-PHP . . . 154

7 Technical Support 159

8 Legal Notice 160

A Appendix 162
A.1 Abbrevations . . . 162
A.2 System Events . . . 164
A.3 Factory Configuration . . . 166
A.4 SNMP VENDOR MIB . . . 167

A.5 SDK Examples 176

List of Figures

5.1 Home 20
5.2 WAN Links 26
5.3 WAN Settings 27
5.4 Link Supervision 28
5.5 Ethernet Ports 29
5.6 Ethernet Link Settings 30
5.7 LAN IP Configuration 32
5.8 SIMs 35
5.9 WWAN Interfaces 38
5.10 WLAN Management 41
5.11 WLAN Scan 43
5.12 WLAN Interfaces 44
5.13 WLAN Configuration 46
5.14 WLAN IP Configuration 47
5.15 USB Device Server 50
5.16 Serial Port 52
5.17 Static Routing 57
5.18 Extended Routing 59
5.19 Multipath Routes 61
5.20 Mobile IP 65
5.21 NAPT Administration 70
5.22 Inbound NAPT 72
5.23 Outbound NAPT 73
5.24 OpenVPN Administration 74
5.25 OpenVPN Configuration 76
5.26 OpenVPN Client Management 78
5.27 IPsec Administration 80
5.28 IPsec Configuration 81
5.29 PPTP Administration 85
5.30 PPTP Tunnel Configuration 86
5.31 PPTP Client Management 87
5.32 Dial-in Server Settings 89
5.33 SDK Administration 95

5.34 SDK Jobs . . . 97
5.35 SDK Testing . . . 99
5.36 DHCP Leases . . . 101
5.37 DHCP Server . . . 103
5.38 DNS Server . . . 104
5.39 NTP Server . . . 105
5.40 Dynamic DNS Settings . . . 106
5.41 E-Mail Settings . . . 108
5.42 Event Notification Settings . . . 110
5.43 SMS Configuration . . . 112
5.44 SSH and Telnet Server . . . 114
5.45 SNMP Agent . . . 118
5.46 Web Server . . . 120
5.47 VRRP Configuration . . . 121
5.48 Voice Gateway . . . 123
5.49 Voice Client Configuration . . . 125
5.50 System . . . 127
5.51 Regional settings . . . 128
5.52 User Accounts . . . 129
5.53 Remote Authentication . . . 131
5.54 Manual File Configuration . . . 134
5.55 Automatic File Configuration . . . 135
5.56 Factory Configuration . . . 136
5.57 Log Viewer . . . 138
5.58 Tech Support File . . . 139
5.59 Keys and certificates management . . . 141
5.60 Licensing . . . 144
5.61 Logout . . . 146

List of Tables

3.1 NB3700 Models . . . 8
3.2 NB3700 Status Indicators . . . 10
3.3 NB3700 Interfaces . . . 11
3.4 USB 2.0 Host Port Specification . . . 12
3.5 Ethernet Port Specification . . . 12
3.6 Pin Assignments of Ethernet Connectors . . . 12

3.7 Power Connector Specifications 13
3.8 Pin Assignments of Power Connector 13
3.9 Isolated Digital Outputs Specification 13
3.10 Isolated Digital Inputs Specification 14
3.11 Pin Assignments of Digital Inputs and Outputs 14
3.12 RS-232 Port Specification 15
3.13 Pin Assignments of RS-232 Port 15

4.1 Operating Conditions 16

5.17 IEEE 802.11 Network Standards 42
5.28 Static Route Flags 58
5.60 SMS Control Commands 100
5.66 SMS Number Expressions 111
5.83 Certificate/Key Terms 142
5.84 Certificate Attributes 142

A.1 Abbreviations 163
A.2 System Events 165
A.3 SDK Examples 178

1. Welcome to NetModule

Thank you for purchasing a NetModule Router. This document should give you an introduction to the router and its features. The following chapters describe any aspects of commissioning the device, installation procedure and provide helpful information towards configuration and maintenance.

2. Conformity

This chapter provides general information for putting the router into operation.

2.1. Safety Instructions

NetModule routers must be used in compliance with any and all applicable national and international laws and with any special restrictions regulating the utilization of the communication module in prescribed applications and environments.

We would like to point out that only the original accessories, shipping with the router, must be used in order to prevent possible injury to health and damage to appliances and to ensure that all the relevant provisions have been complied with. Unauthorized modifications or utilization of unapproved accessories may void the warranty. The routers must not be opened. However, it is possible to replace any pluggable SIM cards even during operation.

All circuits connected to the interfaces of the router must comply with the requirements of Safety Extra Low Voltage (SELV) circuits and have to be designed for indoor use only. Interconnections must not leave the building nor penetrate the body shell of a vehicle. Possible antenna circuits must be limited to over-voltage transient levels below 1500 Volts according to IEC 60950-1, TNV-1 circuit levels using safety approved components.

NB3700 routers shall be only used with certified (CSA or equivalent) power supply, which must have a limited and SELV circuit output. They are basically designed for indoor use. Do not expose the communication module to extreme ambient conditions and protect the communication module against dust, moisture and high temperature.

We remind the user of the duty to observe the restrictions concerning the utilization of radio devices at petrol stations, in chemical facilities or in the course of blasting works in which explosives are used. Switch off the communication module when traveling by plane.

You need to pay heightened attention when using the communication module close to personal medical devices, such as cardiac pacemakers or hearing aids. NetModule routers may also cause interference in the nearer distance of TV sets, radio receivers and personal computers.

Avoid any installation of the antenna during a lightning. Always keep a distance of more than 40 cm from the antenna in order to reduce exposure to electromagnetic fields below the legal limits. This distance applies to $\frac{\lambda}{4}$- and $\frac{\lambda}{2}$-antennas. Larger distances may apply to antennas with higher gain.

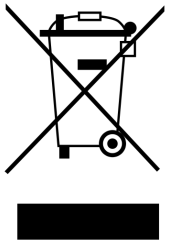
Any Ethernet cabling must be shielded, the Ethernet section of this manual provides

more information.
We highly recommended creating a copy of a working system configuration. It can be downloaded using the Web Manager and easily applied to a newer software release afterwards as we generally guarantee backward compatibility.

2.2. Declaration of Conformity

NetModule hereby declares that under our own responsibility that the routers comply with the relevant standards following the provisions of the *Council Directive 1999/5/EC*. The signed version of the *Declarations of Conformity* can be found on the NetModule web page.

2.3. Waste Disposal



In accordance with the requirements of the *Council Directive 2002/96/EC* regarding Waste Electrical and Electronic Equipment (WEEE), you are urged to ensure that this product will be segregated from other waste at end-of-life and delivered to the WEEE collection system in your country for proper recycling.

2.4. National Restrictions

This product may be generally used in all EU countries (and other countries following the *EU directive 1999/5/EC*) without any limitation except for the countries mentioned below.

2.4.1. France

In case the product is used outdoors, the output power is restricted at some parts of the band. See the table below or check http://www.art-telecom.fr/ for more details.

Frequency	Power (EIRP)	Restrictions
2400-2454 MHz	100 mW (20 dBm)	Only for indoor applications
2454-2483.5 MHz	10 mW (10 dBm)	If used outdoors
5470-5725 MHz		Relevant provisions for the implementation of DFS mechanism described

2.4.2. Italy

This product meets the national radio interface regulations and requirements specified in the *National Frequency Allocation Table* for Italy. Unless operating within the boundaries of the owner's property, the use of this Wireless LAN product requires a general authorization. Please check http://www.comunicazioni.it for more details.

2.4.3. Latvia

The outdoor usage within the 2.4-GHz band requires authorization from the *Electronic Communications Office*. Please check http://www.esd.lv for more details.

2.4.4. Luxembourg

General authorization required for network and service apply.

2.4.5. Norway

Frequency	Restrictions
2400.0-2483.5 MHz	This band range cannot be operated in any geographical areas within a radius of 20km away from the center of Ny-Ålesund

2.4.6. Russian Federation

Frequency	Power (EIRP)	Restrictions
2400.0-2483.5 MHz	100 mW (20 dBm)	Only for indoor applications
5150-5250 MHz	100 mW (20 dBm)	Permitted to use only for indoor applications, closed industrial/warehouse areas and on board of aircrafts
5250-5350 MHz	100 mW (20 dBm)	1. Permitted to use for local networks of crew service communications on board of aircrafts in the area of the airport and at all stages of the flight. 2. Permitted to use for public wireless access local networks on board of a aircraft during the flight but at a altitude of not less than 3000 m

Frequency	Power (EIRP)	Restrictions
5650-5825 MHz	100 mW (20 dBm)	Permitted to use on board of the aircraft during a flight at a altitude not less than 3000 m

2.4.7. Turkey

Frequency	Restrictions
5470-5725 MHz	Not implemented

3. Specifications

3.1. Features

There are several different models of NB3700 available:

Model	UMTS	LTE	WLAN
NB3700-R (Wireline)			
NB3700-W (WLAN)			●
NB3700-U (UMTS)	●		
NB3700-UW (UMTS & WLAN)	●		●
NB3700-2UW (UMTS & WLAN)	2x		●
NB3700-L (LTE)	●	●	
NB3700-LW (LTE & WLAN)	●	●	●
NB3700-2LW (LTE & WLAN)	2x	2x	●

Table 3.1.: NB3700 Models

Note: All UMTS models include support for EDGE/GPRS. All LTE models include support for UMTS/EDGE/GPRS. The UMTS/LTE models can be equipped with a supplementary VOICE (-V) or GNSS (-G) option. We also offer models for CDMA 450MHz (-Ca).

All models have following basic functionality in common:

- Galvanically isolated power supply
- 5 Ethernet ports (M12)
- 1 USB 2.0 host port
- 2 digital inputs
- 2 digital outputs
- 2 SIM card slots

Due to its modular approach, the NB3700 router and its hardware components can be arbitrarily assembled according to its indented usage or application. Please contact us in case of special project requirements.

3.2. Operating Elements

The following table describes the NB3700 status indicators. The color of the LED represents the signal quality for wireless links.

- red means low
- yellow means moderate
- green means good or excellent

Label	Color	State	Function
Status	●	blinking	The device is busy due to startup, software or configuration update.
	●	on	The device is ready. The captions of the top bank apply.
	●	on	The device is ready. The captions of the bottom bank apply.
Mob1	●●●	on	Mobile connection 1 is up.
	●	blinking	Mobile connection 1 is being established.
	❍	off	Mobile connection 1 is down.
Mob2	●●●	on	Mobile connection 2 is up.
	●	blinking	Mobile connection 2 is being established.
	❍	off	Mobile connection 2 is down.
VPN	●	on	VPN connection is up.
	❍	off	VPN connection is down.
WLAN	●●●	on	WLAN connection is up.
	●	blinking	WLAN connection is being established.
	❍	off	WLAN connection is down.
GPS	●	on	GPS is turned on and a valid NMEA stream is available.
	❍	off	GPS is turned off or no valid NMEA stream is available.
Voice	●	on	A voice call is currently active.
	❍	off	No voice call is active.
DO1	●	on	Normally open output port 1 is closed.
	❍	off	Normally open output port 1 is open.

Label	Color	State	Function
DO2	●	on	Normally closed output port 2 is closed.
	❍	off	Normally closed output port 2 is open.
DI1	●	on	Input port 1 is set.
	❍	off	Input port 1 is not set.
DI2	●	on	Input port 2 is set.
	❍	off	Input port 2 is not set.

Table 3.2.: NB3700 Status Indicators

3.3. Interfaces

3.3.1. Overview

Label	Panel	Function
SIM 1	Front	SIM 1, it can be assigned dynamically to any modem by configuration.
SIM 2	Front	SIM 2, it can be assigned dynamically to any modem by configuration.
USB	Front	USB 2.0 host port, can be used as USB device server or for software/configuration updates.
Ethernet 1-4	Front	Ethernet switch ports, can be used as LAN or WAN interface.
Ethernet 5	Front	Additional Ethernet switch port (represents an RS232 port for NB3700-4ES)
⏚	Front	Earth protection connector (earthing is optional), connected to the ground of the power supply V_{GND}. If used, connect a yellow-green marked cable with at least $6mm^2$ copper area. Avoid corrosion and protect the screws against loosening
Power	Front	Power supply (galvanically isolated)
Digital I/O	Front	Galvanically isolated digital I/O M12 connector
Mob 1	Front	TNC female connector for mobile antenna 1
WLAN 1	Front	TNC female connector for first WLAN antenna (main)
GPS	Front	TNC female connector for GPS antenna
Mob 2	Front	TNC female connector for mobile antenna 2
WLAN 2	Front	TNC female connector for second WLAN antenna (diversity)
Reset	Front	Reset button, press at least 3 seconds for reboot and at least 5 second for a factory reset. The start of the factory reset is confirmed by all LEDs lighting up for a second. The button can be released then again.

Table 3.3.: NB3700 Interfaces

3.3.2. USB 2.0 Host Port

The USB 2.0 host port has the following specification:

Feature	Specification
Speed	Low, Full & Hi-Speed
Current	max. 500 mA

Table 3.4.: USB 2.0 Host Port Specification

3.3.3. M12 Ethernet Connectors

Specification

The five Ethernet ports have the following specification:

Feature	Specification
Isolation	1500 Vrms
Speed	10/100 Mbps
Mode	Half- & Full-Duplex
Crossover	Automatic MDI/MDI-X

Table 3.5.: Ethernet Port Specification

Pin Assignment on M12, 4 poles, D-coded female

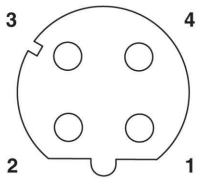
Pin	Signal	Pinning
1	Tx+	
2	Rx+	
3	Tx−	
4	Rx−	

Table 3.6.: Pin Assignments of Ethernet Connectors

3.3.4. Power

The power connector has the following specifications:

Feature	Specification
Power supply nominal voltages	24 V_{DC}, 36 V_{DC} and 48 V_{DC} according to EN 50155
Voltage range	12 V_{DC} to 60 V_{DC} (−15% / +5%)
Max. power consumption	15 W
DC isolation	yes
Power Interruption Class S2	Sustains interruptions up to 10 ms, there are no batteries included
Connector type	

Table 3.7.: Power Connector Specifications

Pin Assignment M12, 4 poles, A-coded male

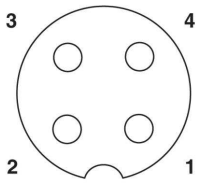
Pin	Signal	Pinning
1	V+ (12−60 V_{DC})	
2	Not connected	
3	V_{GND}	
4	Not connected	

Table 3.8.: Pin Assignments of Power Connector

3.3.5. Digital Inputs and Outputs

Isolated Outputs

The isolated digital output ports have the following specification:

Feature	Specification
Number of output ports	2
Limiting continuous current	1 A
Maximum switching voltage	60 V_{DC}, 42 V_{AC} (V_{rms})
Maximum switching capacity	60 W

Table 3.9.: Isolated Digital Outputs Specification

Isolated Inputs

The isolated digital input ports have the following specification:

Feature	Specification
Number of inputs	2
maximum input voltage	40 V_{DC}
Minimum voltage for level 1 (set)	7.2 V_{DC}
Maximum voltage for level 0 (not set)	5.0 V_{DC}

Table 3.10.: Isolated Digital Inputs Specification

Note: A negative input voltage is not recognized.

Pin Assignment

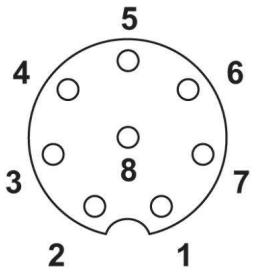
Pin	Signal	Pinning
1	DI1+	
2	DI1−	
3	DI2+	
4	DI2−	
5	DO1: Dry contact relay normally open	
6	DO1: Dry contact relay normally open	
7	DO2: Dry contact relay normally closed	
8	DO2: Dry contact relay normally closed	

Table 3.11.: Pin Assignments of Digital Inputs and Outputs

3.3.6. RS-232 Port

The RS-232 port (if present) has the following specification:

Feature	Specification
Protocol	3-wire RS-232 (TXD, RXD, GND)

Feature	Specification
Baud rate	300, 1 200, 2 400, 4 800, 9 600, 19 200, 38 400, 57 600, 115 200
Data bits	7 bit, 8 bit
Parity	none, odd, even
Stop bits	1, 2
Software flow control	None, XON/XOFF
Hardware flow control	None

Table 3.12.: RS-232 Port Specification

Pin Assignment M12, 4 poles, D-coded female

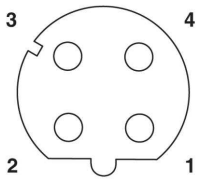
Pin	Signal	Pinning
1	GND	
2	RxD	
3	not connected	
4	TxD	

Table 3.13.: Pin Assignments of RS-232 Port

4. Installation

4.1. Environmental Conditions

The following precautions must be taken before installing a NB3700 router:

- Avoid direct solar radiation
- Protect the device from humidity, steam and aggressive fluids
- Guarantee sufficient circulation of air around the device
- The device is for indoor use only

Parameter	Rating
Input Voltage	12 V_{DC} to 48 V_{DC} (−15% / +20%)
Operating Temperature Range	−25 °C to +70 °C
Humidity	0 to 95% (non-condensing)
Altitude	up to 4000m
Over-Voltage Category	II
Pollution Degree	2
Ingress Protection Rating	IP40 (with SIM and USB covers mounted)

Table 4.1.: Operating Conditions

4.2. Installation of the Router

NB3700 is designed for mounting it on a worktop or wall. Please consider the safety instructions and the environmental conditions in chapter 2.

4.3. Installation of the SIM Card

SIM cards can be inserted by sliding it into one of the designated holes on the front panel. By using a small paper clip (or similar) you will need to press it a bit until it snaps into place. For removing the SIM, you will need to push it again in the same manner. The SIM card will then rebounce and can be pulled out.

SIMs can be assigned flexibly to any modem in the system. It is also possible to switch a SIM to a different modem during operation, for instance if you want to use another

provider upon a certain condition. However, a SIM switch usually takes about 10-20 seconds which can be bypassed (e.g. at bootup) if SIMs are installed reasonably. Using only a single SIM with one modem, it should be preferably placed into the SIM 1 holder. For systems which should operate two modems with two SIMs in parallel, we recommend to assign **Mobile 1** to SIM 1 and **Mobile 2** to SIM 2.
Further information about SIM configuration can be found in chapter 5.3.3.

4.4. Installation of the WLAN Antennas

Any WLAN antennas must be mounted to the connectors **WLAN1** and **WLAN2**. The number of attached antennas can be configured in the software. If only one antenna is used, it must be attached to **WLAN1**. However, for better diversity and thus better throughput and coverage, we highly recommend using two antennas.

4.5. Installation of the Local Area Network

Up to two 10/100 Mbps Ethernet devices can be directly connected to the router, further devices can be attached via an addtional Ethernet switch. Please ensure that the connector has been plugged in properly and remains in a fixed state, you might otherwise experience sporadical link loss during operation. The Link/Act LED will lit up as soon as the device has synced. If not, it might be necessary to configure a different link setting as described in chapter **??**.

4.6. Installation of the Power Supply

The router can be powered with an external source supplying between 12 V_{DC} and 48 V_{DC}. It is to be used with a certified (CE or equivalent) power supply, which must have a limited and SELV circuit output. The router is now ready for getting engaged.

5. Configuration

The following chapters give information about setting up the router and configuring its features as provided with system software 3.7.

5.1. First Steps

NetModule routers can be easily set up by using the HTTP-based configuration interface, called the Web Manager. It is supported by the latest web browsers (e.g. Microsoft Internet Explorer 11, Mozilla Firefox 28.0, Safari 7 and many others). Please ensure to have JavaScript turned on.

Any submitted configuration via the Web Manager will be applied immediately to the system when pressing the `Apply` button. When configuring subsystem like WLAN which requires multiple steps, you may use the `Continue` button to store any settings temporarily and apply them at a later time. Please note, that those settings will be neglected at logout unless applied.

You may also upload configuration files via SNMP, SSH, HTTP or USB in case you intend to deploy a larger numbers of routers. Advanced users may also use the Command Line Interface (CLI) and set configuration parameters directly.

The IP address of Ethernet1 is `192.168.1.1` and the Dynamic Host Configuration Protocol (DHCP) is activated on the interface by default. The following steps need to be taken to establish your first Web Manager session:

1. Connect the Ethernet port of your computer to the Ethernet1 port of the router using a standard CAT5 cable with RJ45 (or M12) connectors.
2. If not yet activated, enable DHCP on your computer's Ethernet interface so that an IP address can be obtained automatically from the router. This usually takes a short amount of time until your PC has received the corresponding parameters (IP address, subnet mask, default gateway, name server). You may track the progress by having a look to your network control panel and check whether your PC has correctly retrieved an IP address of the range `192.168.1.100` to `192.168.1.199`.
3. Launch your favorite web browser and point it to the IP address of the router (the URL is http://192.168.1.1).
4. Please follow the instructions of the Web Manager for configuring the router. Most of the menus are self-explanatory, further details are given in the following chapters.

5.1.1. Initial Access

In factory state you will be prompted for a new administrator password. Please choose a password which is both, easy to remember but also robust against dictionary attacks (such as one that contains numbers, letters and punctuation characters). The password shall have a minimum length of 6 characters. It shall contain a minimum of 2 numbers and 2 letters.

Please note that the admin password will be also applied for the root user which can be used to access the device via the serial console, telnet, SSH or to enter the bootloader. You may also configure additional users which will only be granted to access the summary page or retrieve status information but not to set any configuration parameters.

A set of services (USB Autorun, CLI-PHP) are by default activated in factory state and will be disabled as soon as the admin password has been set. They can be enabled again afterwards in the relevant sections.

5.1.2. Recovery

Following actions might be taken in case the router has been misconfigured and cannot be reached anymore:

1. Factory Reset: You can initiate a reset back to factory settings via the Web Manager, by running the command `factory-reset` or by pressing the reset button. The latter would require a slim needle or paper clip which must be inserted into the hole to the left of the LEDs . The button must be hold pressed for up to 5 seconds until all LEDs flash up.
2. Serial Console Login: It is also possible to log into the system via the serial port. This would require a terminal emulator (such as PuTTY or HyperTerminal) and an RS232 connection (115200 8N1) attached to the serial port of your local computer.
3. Recovery Image: In severe cases we can provide a recovery image on demand which can be loaded into RAM via TFTP and executed. It offers a minimal system image for running a software update or doing other modifications. You will be provided with two files, recovery-image and recovery-dtb, which must be placed in the root directory of a TFTP server (connected via LAN1 and address 192.168.1.254). The recovery image can be launched from the boot-loader using a serial connection. You will have to stop the boot process by pressing `s` and enter the bootloader. You can then issue `run recovery` to load the image and start the system which can be accessed via HTTP/SSH/Telnet and its IP address `192.168.1.1` afterwards. This procedure can be also initiated by holding the factory reset button longer than 15 seconds.

5.2. HOME

This page provides a status overview of enabled features and connections.

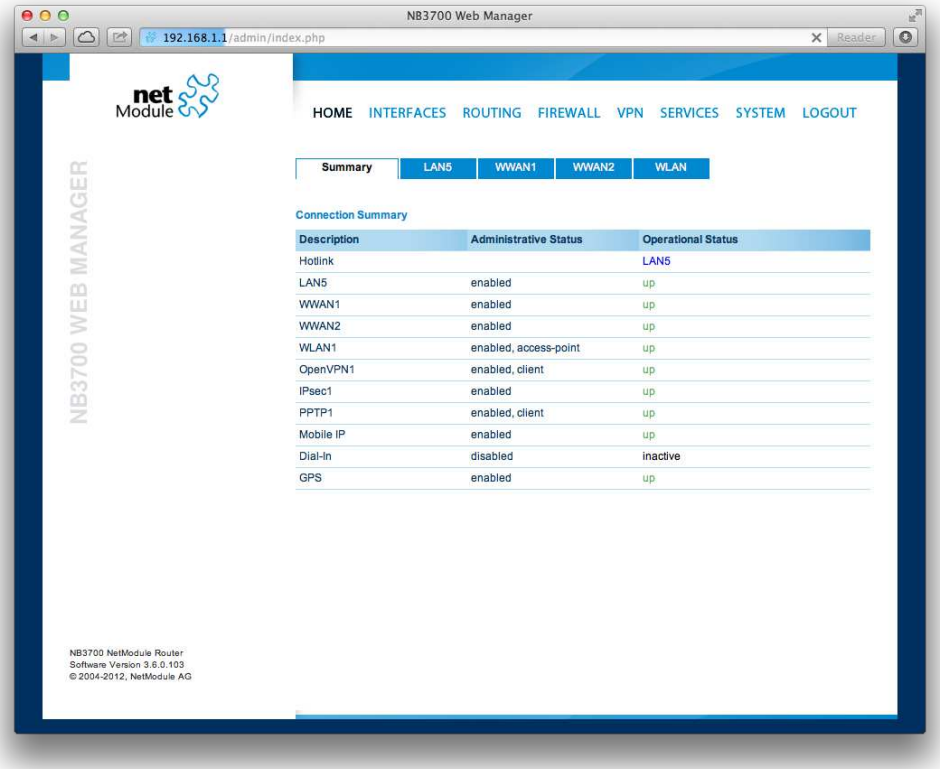

NB3700 Web Manager
192.168.1.1/admin/index.php
Reader
net Module
HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM LOGOUT
NB3700 WEB MANAGER
Summary LAN5 WWAN1 WWAN2 WLAN
Connection Summary
Description | Administrative Status | Operational Status
Hotlink | | LAN5
LAN5 | enabled | up
WWAN1 | enabled | up
WWAN2 | enabled | up
WLAN1 | enabled, access-point | up
OpenVPN1 | enabled, client | up
IPsec1 | enabled | up
PPTP1 | enabled, client | up
Mobile IP | enabled | up
Dial-In | disabled | inactive
GPS | enabled | up
NB3700 NetModule Router
Software Version 3.6.0.103
© 2004-2012, NetModule AG

Figure 5.1.: Home

Summary

This page offers a short summary about the administrative and operational status of the router's interfaces.

WAN

This page offers details about any enabled Wide Area Network (WAN) links (such as the IP addresses, network information, signal strength, etc.) The information about the amount of downloaded/uploaded data is stored in non-volatile memory, thus survive a reboot of the system.
The counters can be reset by pressing the *Reset* button.

WLAN

The WLAN page offers details about the enabled WLAN interfaces when operating in access-point mode. This includes the SSID, IP and MAC address and the currently used frequency and transmit power of the interface as well as the list of associated stations.

GNSS

This page displays the position status values, such as latitude/longitude, the satellites in view and more details about the used satellites.

Ethernet

This page shows information about the Ethernet interfaces and packet statistics information.

LAN

This page shows information about the LAN interfaces plus the neighborhood information.

DHCP

This page offers details about any activated DHCP service, including a list of issued DHCP leases.

OpenVPN

This page provides information about the OpenVPN tunnel status.

IPSec

This page provides information about the IPsec tunnel status.

PPTP

This page provides information about the PPTP tunnel status.

GRE

This page provides information about the GRE tunnel status.

MobileIP

This page provides information about Mobile IP connections.

Firewall

This page offers information about any firewall rules and their matching statistics. It can be used to debug the firewall.

QoS

This page provides information about the used QoS queues.

System Status

The system status page displays various details of your NB3700 router, including system details, information about mounted modules and software release information.

SDK

This section will list all webpages generated by SDK scripts.

5.3. INTERFACES

5.3.1. WAN

Link Management

Depending on your hardware model, WAN links can be made up of either Wireless Wide Area Network (WWAN), Wireless LAN (WLAN), Ethernet or PPP over Ethernet (PPPoE) connections. Please note that each WAN link has to be configured and enabled in order to appear on this page.

Generally, a link will be only dialed or declared as up if the following prerequisites are met:

Condition	WWAN	WLAN	ETH	PPPoE
Modem is registered	X			
Registered with valid service type	X			
Valid SIM state	X			
Sufficient signal strength	X	X		
Client is associated		X		
Client is authenticated		X		
Valid DHCP address retrieved	X	X	X	X
Link is up and holds address	X	X	X	X
Ping check succeeded	X	X	X	X

The menu can be used further to prioritize your WAN links. The highest priority link which has been established successfully will become the so-called **hotlink** which holds the default route for outgoing packets.

In case a link goes down, the system will automatically switch over to the next link in the priority list. You can configure each link to be either established when the switch occurs or permanently in order to minimize link downtime.

Parameter	WAN Link Priorities
1st priority	The primary link which will be used whenever possible.
2nd priority	The first fallback link, it can be enabled permanently or being dialed as soon as Link 1 goes down.

Parameter	WAN Link Priorities
3rd priority	The second fallback link, it can be enabled permanently or being dialed as soon as Link 2 goes down.
4th priority	The third fallback link, it can be enabled permanently or being dialed as soon as Link 3 goes down.

Links are being triggered periodically and put to sleep in case it was not possible to establish them within a certain amount of time. Hence it might happen that permanent links will be dialed in background and replace links with lower priority again as soon as they got established. In case of interfering links sharing the same resources (for instance in dual-SIM operation) you may define a switch-back interval after which an active hotlink is forced to go down in order to let the higher-prio link getting dialed again.

We recommend to use the `permanent` operation mode for WAN links in general. However, in case of time-limited mobile tariffs for instance, the `switchover` mode might be applicable. By using the `distributed` mode, it is possible to distribute outgoing traffic over multiple WAN links based on their weight ratio.

For mobile links, it is further possible to pass-through the WAN address towards a local host (also called Drop-In). In particular, the first DHCP client of the specified interface will receive the public IP address. More or less, the system acts like a modem in such case which can be helpful in case of firewall issues. Once established, the Web Manager can be reached over port 8080 using the public address.

Parameter	WAN Link Operation Modes
disabled	Link is disabled
permanent	Link is being established permanently
on switchover	Link is being established on switchover, it will be dialled if previous links failed
distributed	Link is member of a load distribution group

Parameter	WAN Link Settings
Operation mode	The operation mode of the link
Weight	The weight ratio of a distributed link
Switch-back	Specifies the switch-back condition of a switchover link and the time after an active hotlink will be teared down

Parameter	WAN Link Settings
IP Passthrough	Specifies whether the IP address of a link should be passed-through to the first DHCP client of a LAN interface

Settings

This page can be used to configure WAN specific settings like the Maximum Segment Size (MSS). The MSS corresponds to the largest amount of data (in bytes) that the router can handle in a single, unfragmented TCP segment. In order to avoid any negative side effects the number of bytes in the data segment and the headers must not add up to more than the number of bytes in the Maximum Transmission Unit (MTU). The MTU can be configured per each interface and corresponds to the largest packet size that can be transmitted.

Parameter	TCP MSS Settings
MSS adjustment	Enable or disable MSS adjustment on WAN interfaces.
Maximum segment size	Maximum number of bytes in a TCP data segment.

Supervision

Network outage detection can be performed by sending pings on each link to some authoritative hosts. A link will be declared as down in case all trials have failed and only as up if at least one host can be reached.

Parameter	Supervision Settings
Link	The WAN link to be monitored (can be ANY)
Mode	Specifies whether the link shall only be monitored if being up or if connectivity shall be also validated at connection establishment
Primary host	The primary host to be monitored
Secondary host	The secondary host to be monitored (optional)
Ping timeout	The amount of time in milliseconds a response for a single ping can take, consider to increase this value in case of slow and tardy links (such as 2G connections)
Ping interval	The interval in seconds at which pings are transmitted on each interface

Parameter	Supervision Settings
Retry interval	The interval in seconds at which pings are retransmitted in case a first ping failed
Max. number of failed trials	The maximum number of failed ping trials until the link will be declared as down
Emergency action	The emergency action which should be taken after a maximum downtime has been reached. Using `reboot` would perform a reboot of the system, `restart link services` will restart all link-related applications including a reset of the modem.

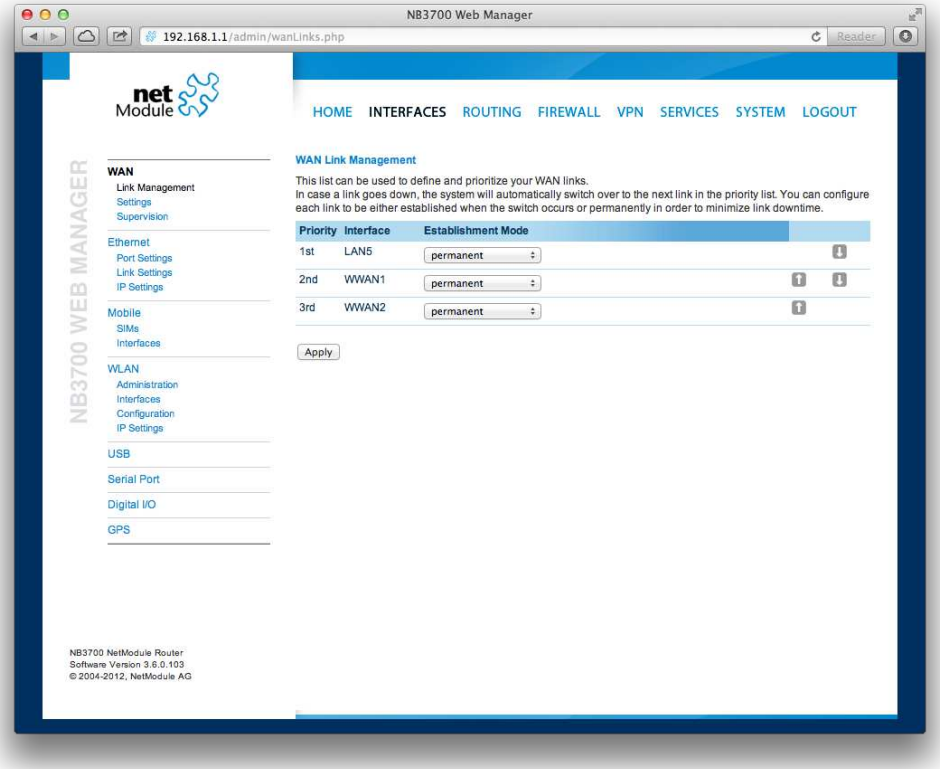


Figure 5.2.: WAN Links

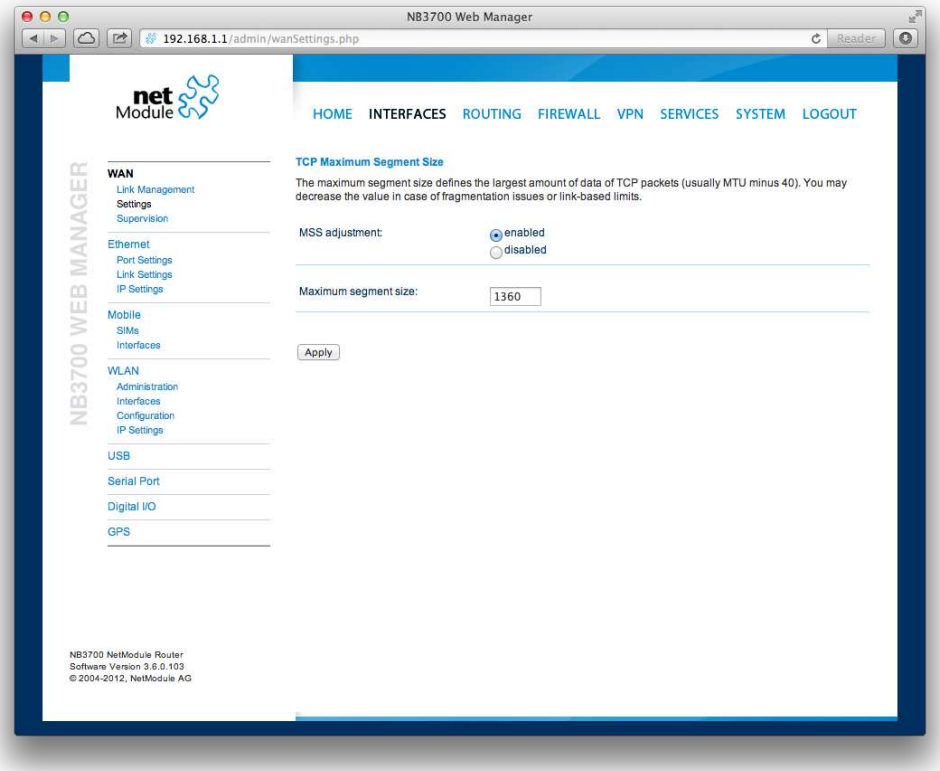


Figure 5.3.: WAN Settings

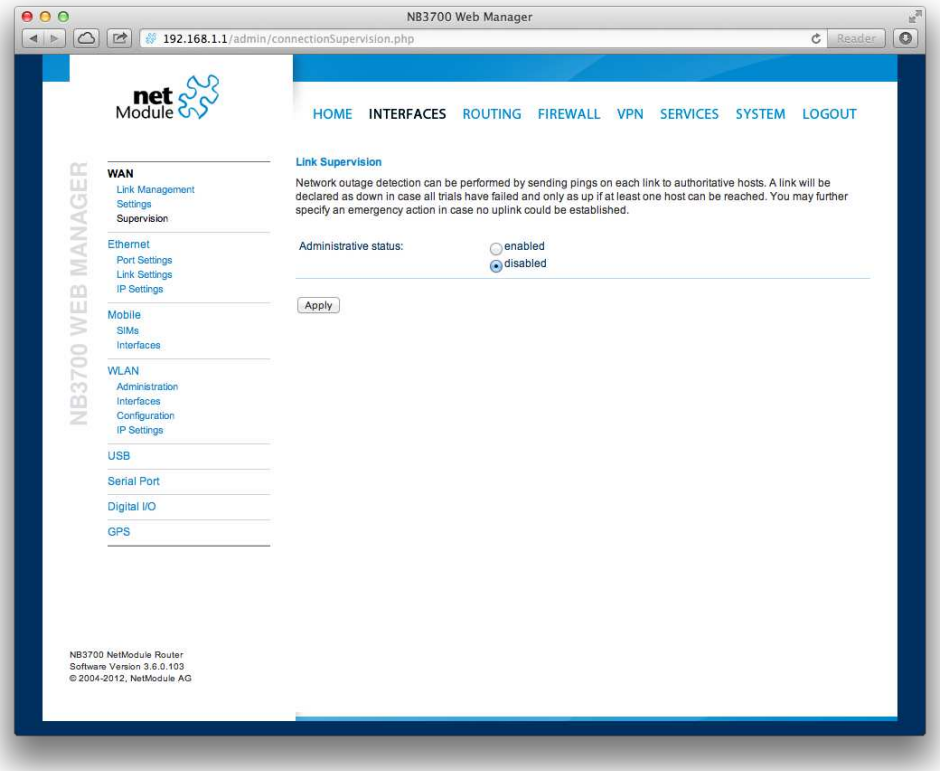


Figure 5.4.: Link Supervision

5.3.2. Ethernet

NB3700 routers ship with an Ethernet switch (ETH1-ETH5) which can be linked via M12 connectors.
ETH1 usually forms the LAN1 interface which should be used for LAN purposes. Other interfaces can be used to connect other LAN segments or for configuring a WAN link.

Port Assignment

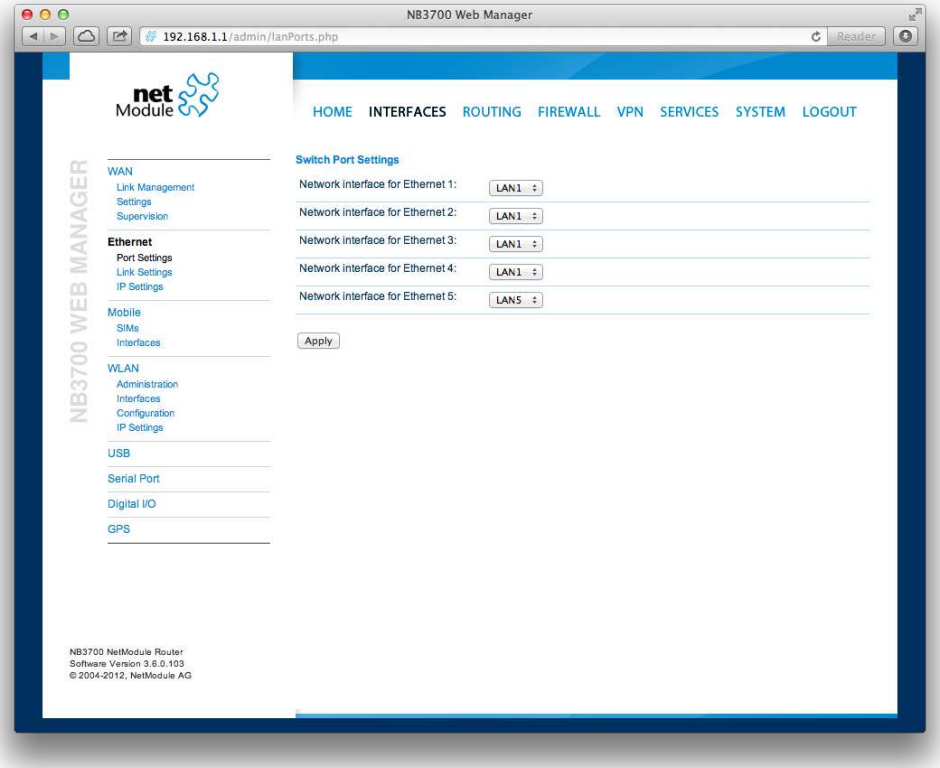


Figure 5.5.: Ethernet Ports

This menu can be used to individually assign each Ethernet port to a LAN interface, just in case you want to have different subnets per port or use one port as WAN interface. You may assign multiple ports to the same interface. Please note that on systems without an Ethernet switch, the ports will be bridged by software then and operated by running the Spanning Tree Protocol (STP).

Link Settings

Link negotiation can be set for each Ethernet port individually. Most devices support auto-negotiation which will configure the link speed automatically to comply with other devices in the network. In case of negotiation problems, you may assign the modes

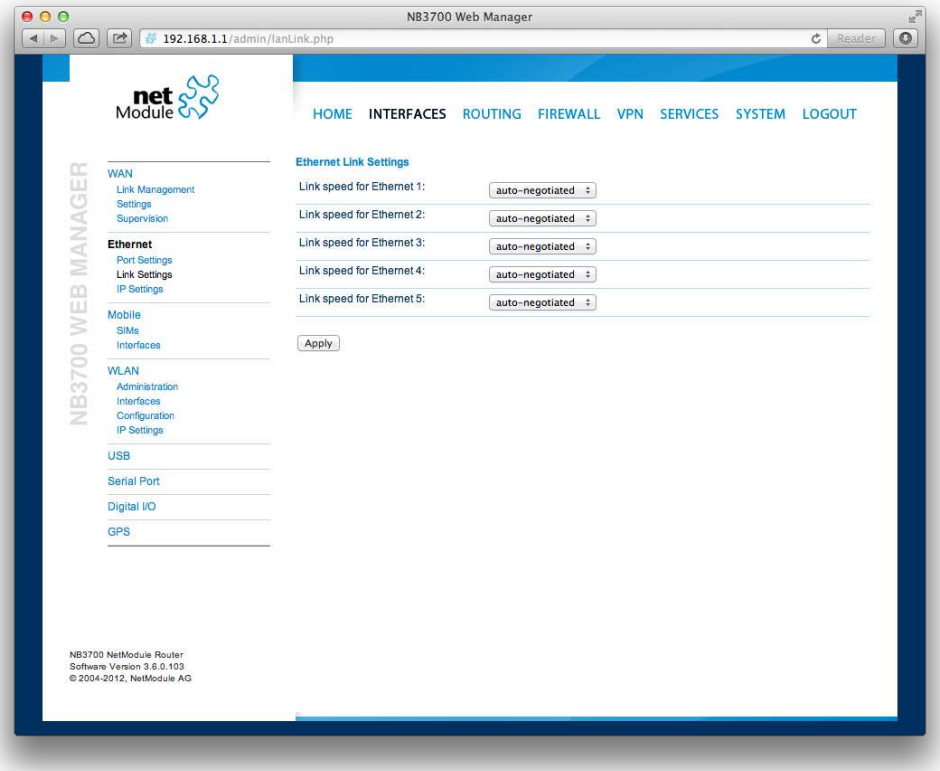


Figure 5.6.: Ethernet Link Settings

manually but it has to be ensured that all devices in the network utilize the same settings then.

VLAN Management

NetModule routers support Virtual LAN according to IEEE 802.1Q which can be used to create virtual interfaces on top of an Ethernet interface. The VLAN protocol inserts an additional header to Ethernet frames carrying a VLAN Identifier (VLAN ID) which is used for distributing the packets to the associated virtual interface. Any untagged packets, as well as packets with an unassigned ID, will be distributed to the native interface. In order to form a distinctive subnet, the network interface of a remote LAN host must be configured with the same VLAN ID as defined on the router. Further, 802.1P introduces a priority field which influences packet scheduling in the TCP/IP stack.

The following priority levels (from lowest to highest) exists:

Parameter	VLAN Priority Levels
0	Background
1	Best Effort
2	Excellent Effort
3	Critical Applications
4	Video (< 100 ms latency and jitter)
5	Voice (< 10 ms latency and jitter)
6	Internetwork Control
7	Network Control

IP Settings

This page can be used to configure IP addressing for your LAN/WAN Ethernet interfaces. In addition to the primary IP address/subnet mask you may define an additional IP address alias on the interface.

Please keep in mind that the DNS servers can be set globally in the DNS server configuration menu. But as soon as a link comes up it will use the interface-specific nameservers (e.g. the ones being retrieved over DHCP) and update the resolver configuration accordingly.

Parameter	LAN IP Settings
Mode	Defines whether this interface is being used as LAN or WAN interface

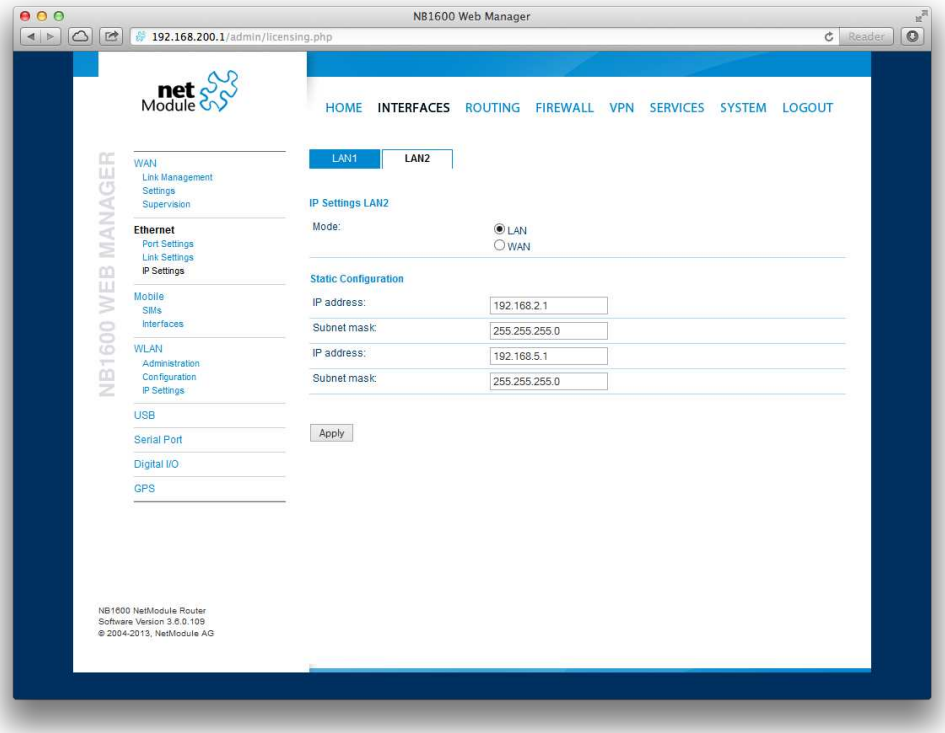


Figure 5.7.: LAN IP Configuration

When running in LAN mode, the interface may be configured with the following settings:

Parameter	LAN IP Settings
IP address	The IP interface address
Subnet mask	The subnet mask for this interface
Alias IP address	The alias IP interface address
Alias subnet mask	The alias subnet mask for this interface

When running in WAN mode, the interface may be configured with the following settings:

Parameter	WAN IP Settings
WAN mode	The WAN operation mode, defines whether the interface should run as DHCP client, statically configured or over PPPoE.
MTU	The maximum transfer unit for the interface, if provided it will specify the largest size of a packet transmitted on the interface.

When running as DHCP client, no further configuration is required because all IP-related settings (address, subnet, gateway, DNS server) will be retrieved from a DHCP server in the network. You may also define static values but caution has to be taken to assign an unique IP address as it would otherwise raise IP conflicts in the network.

PPPoE is commonly used when communicating with another WAN access device (like a DSL modem). The following settings can be applied:

Parameter	PPPoE Configuration
User name	PPPoE user name for authenticating at the access device
Password	PPPoE password for authenticating at the access device
Service name	Specifies the service name set of the access concentrator and can be left blank unless you have multiple services on the same physical network and need to specify the one you want to connect to.

Parameter	PPPoE Configuration
Access concentrator name	The name of the concentrator (the PPPoE client will connect to any access concentrator if left blank)

5.3.3. Mobile

SIMs

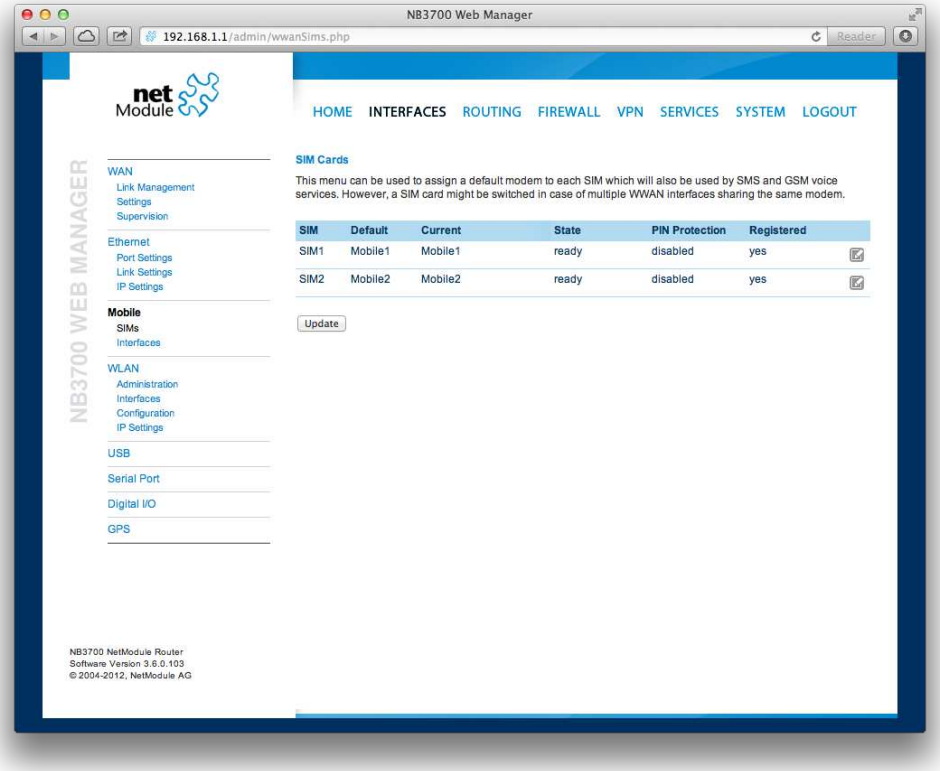


Figure 5.8.: SIMs

The SIM page gives an overview about the available SIM cards, their assigned modems and the current state. Once a SIM card has been inserted, assigned to a modem and successfully unlocked, the card should remain in state `ready` and the network registration status should have turned to `registered`. If not, please double-check your PIN.

Please keep in mind that registering to a network usually takes some time and depends on signal strength and possible radio interferences. You may hit the `Update` button at any time in order to restart PIN unlocking and trigger another network registration attempt.

Under some circumstances (e.g. in case the modem flaps between base stations) it might be necessary to set a specific service type or assign a fixed operator. The list of operators around can be obtained by initiating a network scan (may take up to 60 seconds). Further details can be retrieved by querying the modem directly, a set of suitable commands can be provided on request.

Configuration

A SIM card is generally assigned to a default modem but might be switched, for instance if you set up two WWAN interfaces with one modem but different SIM cards.
Close attention has to be paid when other services (such as SMS or Voice) are operating on that modem, as a SIM switch will naturally affect their operation.
The following settings can be applied:

Parameter	WWAN SIM Configuration
Default modem	The default modem assigned to this SIM card
Service type	The service type to be used by default with this SIM card. Remember that the link manager might change this in case of different settings. The default is to use `automatic`, in areas with interfering base stations you can force a specific type (e.g. 3G-only) in order to prevent any flapping between the stations around.
PIN protection	Depending on the used card, it can be necessary to unlock the SIM with a PIN code. Please check the account details associated with your purchased SIM and figure out whether it is protected with a PIN.
PIN code	The PIN code for unlocking the SIM card
SMS gateway	The service center number for sending short messages. It is generally retrieved automatically from your SIM card but you may define a fix number here.

Network

This page provides information about the current network status, signal strength and the Local Area Identifier (LAI) to which the modem has been registered. An LAI is a globally unique number that identifies the country, network provider and Local Area Code (LAC, group of base stations) of any given location area. It can be used to force the modem to register to a particular mobile cell in case of competing stations.
You may further initiate a mobile network scan for getting networks in range and assign an LAI manually.

Query

This page allows you to send Hayes AT commands to the modem. Besides the 3GPP-conforming AT command-set further modem-specific commands can be applicable which we can provide on demand. Some modems also support running Unstructured Supplementary Service Data (USSD) requests, e.g. for querying the available balance of a prepaid account.

WWAN Interfaces

This page can be used to manage your WWAN interfaces. The resulting link will pop up automatically as WAN link once an interface has been added. Please refer to chapter 5.3.1 for how to manage them.

The Mobile LED will be blinking during the connection establishment process and goes on as soon as the connection is up. Refer to section 5.8.5 or consult the system log files for troubleshooting the problem in case the connection did not come up.

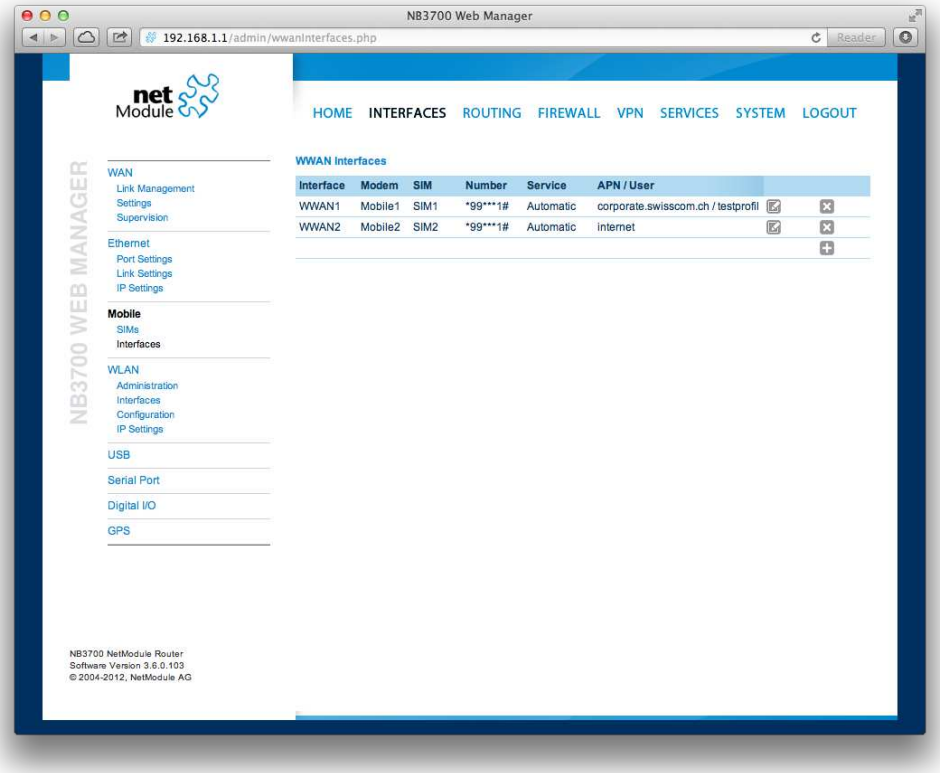


Figure 5.9.: WWAN Interfaces

The following mobile settings are required:

Parameter	WWAN Mobile Parameters
Modem	The modem to be used for this WWAN interface
SIM	The SIM card to be used for this WWAN interface
Service type	The required service type

Please note that these settings supersede the general SIM based settings as soon as the link is being dialed.

Generally, the connection settings are derived automatically as soon as the modem has registered and the network provider has been found in our database. Otherwise, it will be required to configure the following settings manually:

Parameter	WWAN Connection Parameters
Phone number	The phone number to be dialed, for 3G+ connections this commonly refers to be *99***1#. For circuit-switched 2G connections you can enter the fixed phone number to be dialed in international format (e.g. +41xx).
Access point name	The access point name (APN) being used
Authentication	The authentication scheme being used, if required this can be PAP or/and CHAP
Username	The user-name used for authentication
Password	The password used for authentication

Furtheron, you may configure the following advanced settings:

Parameter	WAN Advanced Parameters
Required signal strength	Sets a minimum required signal strength before the connection is dialed
Home network only	Determines whether the connection should only be dialed when registered to a home network
Negotiate DNS	Specifies whether the DNS negotiation should be performed and the retrieved name-servers should be applied to the system
Call to ISDN	Has to be enabled in case of 2G connections talking to an ISDN modem
Header compression	Enables or disables 3GPP header compression which may improve TCP/IP performance over slow serial links. Has to be supported by your provider.
Data compression	Enables or disables 3GPP data compression which shrinks the size of packets to improve throughput. Has to be supported by your provider.
Client address	Specifies a fixed client IP address if assigned by the provider

Parameter	WAN Advanced Parameters
MTU	The Maximum Transmission Unit for this interface

5.3.4. WLAN

WLAN Management

In case your router is shipping with a WLAN (or Wi-Fi) module you can operate it either as `client` or `access point`. As a `client` it can create an additional WAN link which for instance can be used as backup link. As access point, it can form another LAN interface which can be either bridged to an Ethernet-based LAN interface or create a self-contained IP interface which can be used for routing and to provide services (such as DHCP/DNS/NTP) in the same way like an Ethernet LAN interface does.

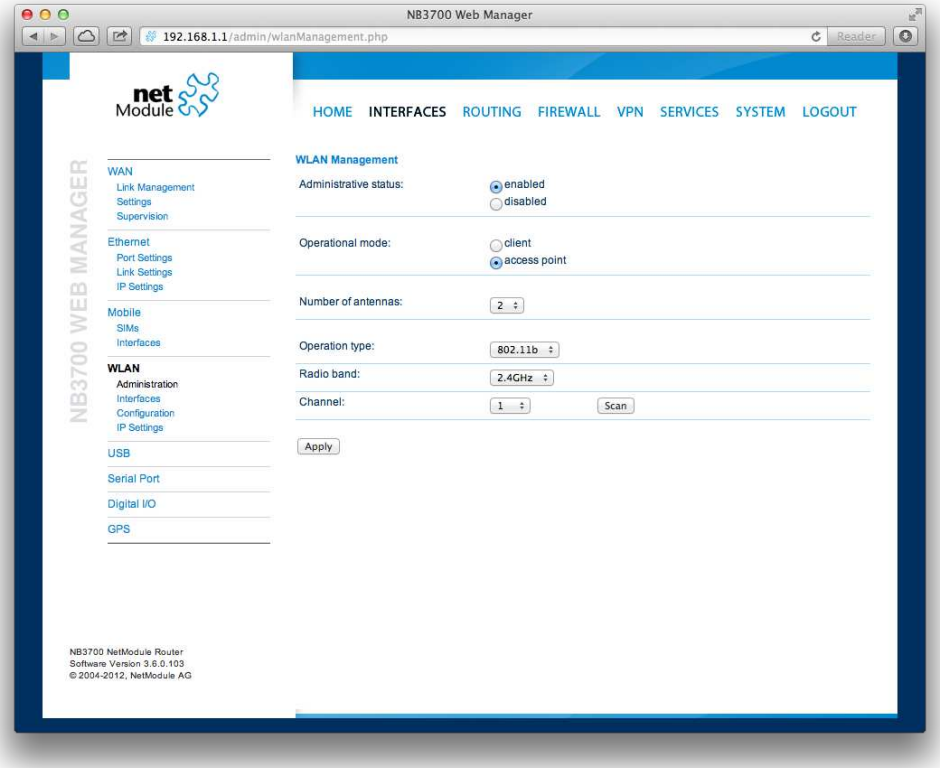


Figure 5.10.: WLAN Management

If the administrative status is set to `disabled`, the module will be powered off in order to reduce the overall power consumption. Regarding antennas, we generally recommend using two antennas for better coverage and throughput. A second antenna is definitely mandatory if you want to achieve higher throughput rates in 802.11n.

A WLAN `client` will automatically became a WAN link and can be managed as described in chapter 5.3.1.

Running as access point, you can further configure the following settings:

Parameter	WLAN Management
Operation type	Specifies the desired IEEE 802.11 operation mode
Radio band	Selects the radio band to be used for connections, depending on your module it could be 2.4 or 5 GHz
Channel	Specifies the channel to be used

Available operation modes are:

Standard	Frequencies	Bandwidth	Net Data Rate	Range Indoor/Outdoor
802.11a	5 GHz	20 MHz	54 Mbit/s	35m / 120m
802.11b	2.4 GHz	20 MHz	11 Mbit/s	35m / 140m
802.11g	2.4 GHz	20 MHz	54 Mbit/s	38m / 140m
802.11n	2.4/5 GHz	20/40 MHz	150 Mbit/s	70m / 250m

Table 5.17.: IEEE 802.11 Network Standards

Prior to setting up an access point, it is always a good idea to run a network scan for getting a list of neighboring WLAN networks and then choose the less interfering channel. Please note that two adequate channels are required for getting good throughputs with 802.11n and a bandwidth of 40 MHz.

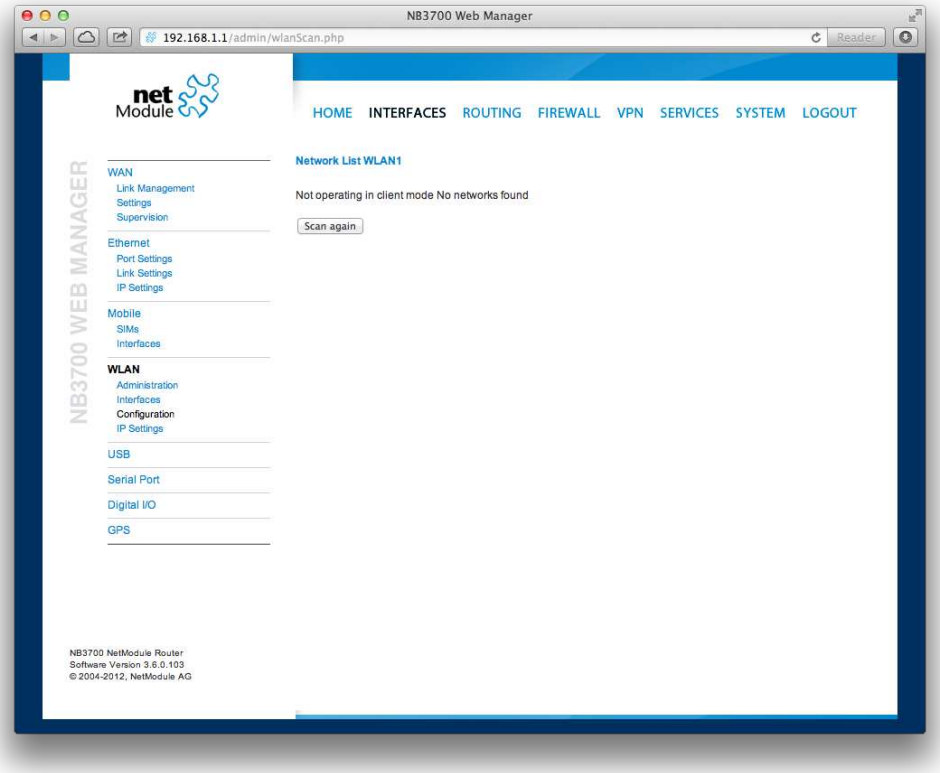


Figure 5.11.: WLAN Scan

Running in `client` mode, you can select the network to which you want to connect to and enter the required authentication settings. You may also perform a WLAN network scan and pick the settings from the discovered information directly. The credentials can be obtained by the operator of your WLAN access point.

WLAN Interfaces

An access point can define up to 4 networks being broadcasted. The networks can be individually bridged to a LAN interface or operate as dedicated interface in routing-mode.

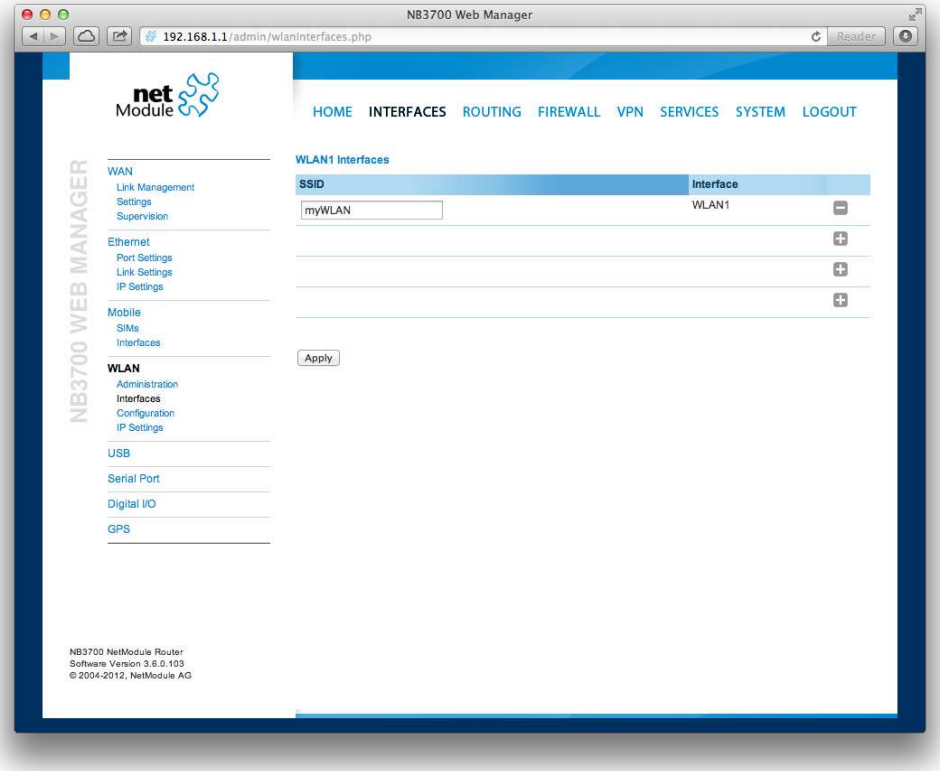


Figure 5.12.: WLAN Interfaces

WLAN Configuration

Running in access point mode you can define up to 4 SSIDs with each running their own network configuration. This section can be used to configure security-related settings.

Parameter	WLAN Configuration
SSID	The network name (called SSID)

Parameter	WLAN Configuration
Security mode	The desired security mode. WPA-PSK provides password-based authentication, WPA-RADIUS can be used to authenticate against a remote RADIUS server which can be configured in chapter 5.8.2 and WPA-EAP-TLS performs authentication using keys/certificates which can be configured in chapter 5.8.6.
WPA/WPA2 mixed mode	WPA2 should be preferred over WPA1, running WPA/WPA2 mixed-mode offers both.
WPA cipher	The WPA cipher to be used, the default is to run both (TKIP and CCMP)
Passphrase	The passphrase used for authentication with WPA-PSK, otherwise the key passphrase for WPA-EAP-TLS
Identity	The identity used for WPA-RADIUS and WPA-EAP-TLS

Being a shared medium, we strongly advise to secure your WLAN connection using passwords or even keys/certificates.

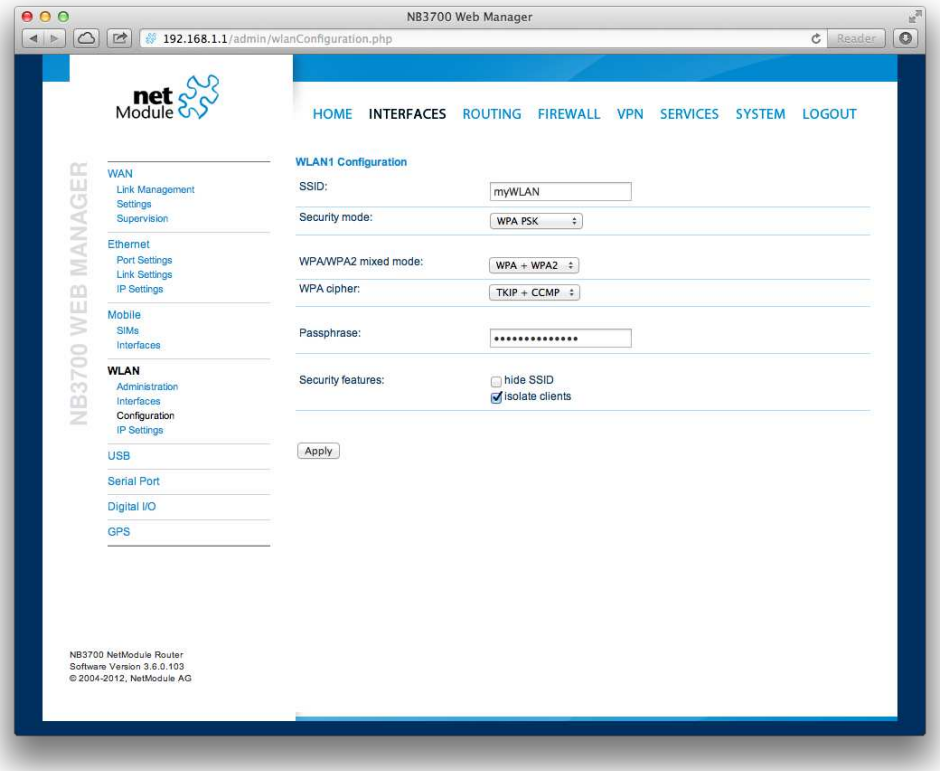


Figure 5.13.: WLAN Configuration

WLAN IP Settings

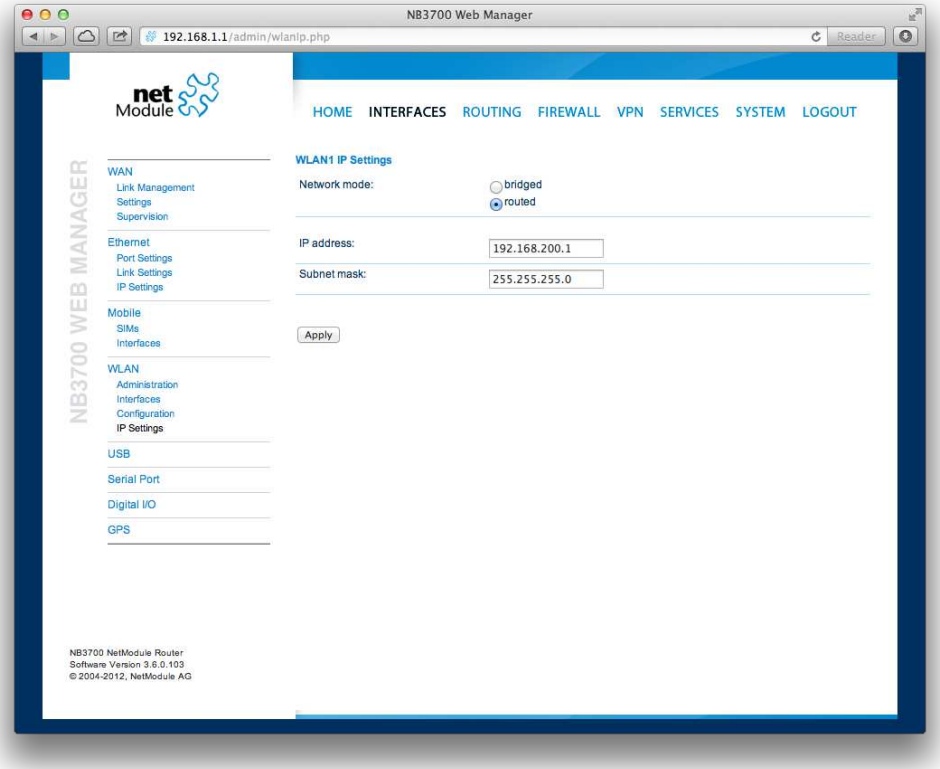


Figure 5.14.: WLAN IP Configuration

This section lets you configure the TCP/IP settings of your WLAN network.

A `client` interface can be run over DHCP or with a statically configured address and default gateway.

The access point networks can be bridged to any LAN interface for letting WLAN clients and Ethernet hosts operate in the same subnet. However, for multiple SSIDs we strongly recommend to set up separated interfaces in routing-mode in order to avoid unwanted access and traffic between the interfaces. The corresponding DHCP server for each network can be configured in afterwards as described in chapter 5.7.2.

Parameter	WLAN IP Settings
Network mode	Choose whether the interface shall be operated bridged or in routing-mode
Bridge interface	If bridged, the LAN interface to which the WLAN network should be bridged

Parameter	WLAN IP Settings
IP address / netmask	In routing-mode, the IP address and netmask for this WLAN network

5.3.5. USB

NetModule routers ship with a standard USB host port which can be used to connect a storage, network or serial USB device. Please contact our support in order to get a list of supported devices.

USB Administration

Parameter	USB Administration
Administrative status	Specifies whether devices shall be recognized
Enable hotplug	Specifies whether device shall be recognized if plugged in during runtime or only at bootup
Enable USB/IP device server	Specifies if devices shall be exported over IP

If the USB/IP device server has been enabled you can discover the mounted USB devices and attach them to the USB/IP server. Enabled devices can now be exported to a remote host. You will need an additional driver on the client for which we provide Windows or Linux drivers. Further installation instructions can be provided on demand.

Please note that some USB devices behave latency-sensitive which may raise problems when operating over a slow IP connection. Some devices may generally not work with the USB/IP driver. Please contact our support in case of compatibility issues.

USB Devices

This page show the currently connected devices and it can be used to enable a specific device based on its Vendor and Product ID.

Parameter	USB Devices
Vendor ID	The USB Vendor ID of the device
Product ID	The USB Product ID of the device
Module	The USB module to be applied for this device

Any ID must be specified in hexadecimal notation, wildcards are supported (e.g. `AB[0-1][2-3]` or `AB*`)

USB Autorun

This feature can be used to automatically launch a shell script or perform a software/-config update as soon as an USB storage stick has been plugged in. For authentication, a file called `autorun.key` must exist in the root directory of a FAT16/32 formatted stick. It can be downloaded from that page and holds the SHA256 hash key of the admin

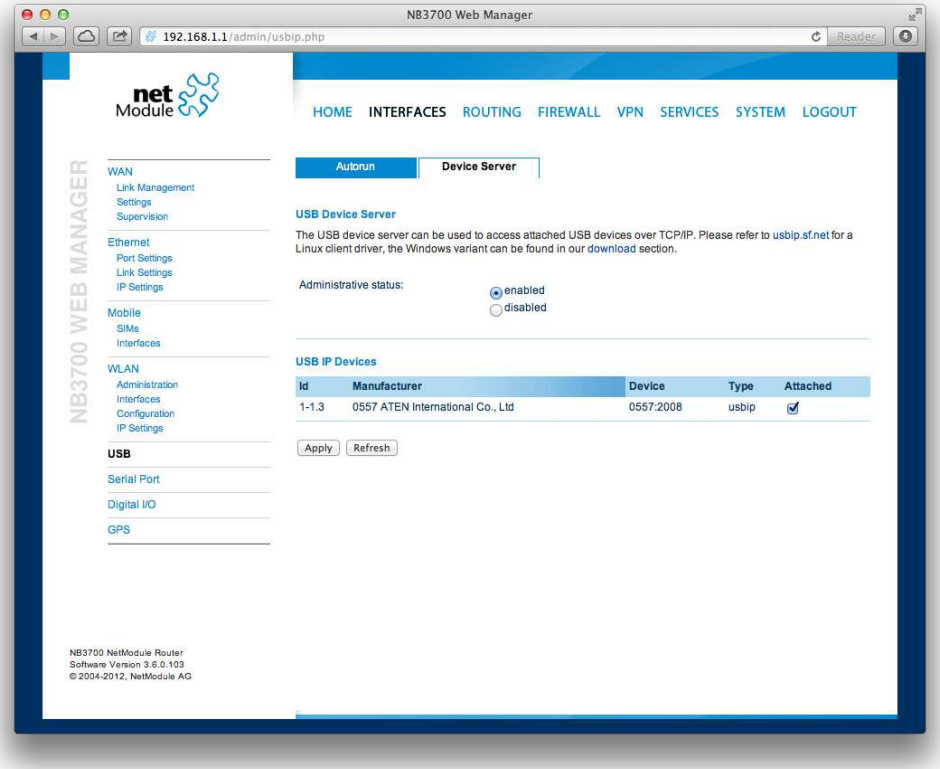


Figure 5.15.: USB Device Server

password. The file can hold multiple hashes which will be processed line-by-line during authentication which can be used for setting up more systems with different admin passwords.

For new devices with an empty password the hash key

`e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855`

can be used.

The hash keys can be generated by running the command `echo -n "<admin-password>" | sha256sum` on a Linux system.

Once authentication has succeeded, the system scans for other files in the root directory which can perform the following actions:

1. For running a script: `autorun.sh`
2. For a configuration update: `cfg-<SERIALNO>.zip` (e.g. `cfg-00112B000815.zip`), or if not available `cfg.zip`
3. For a software update: `sw-update.img`

5.3.6. Serial Port

This page can be used to manage your serial ports. They can be used for various purposes on the system. When set to `none` it will be disabled, when set to `login console` you would be able to get a login shell when connecting to the serial port (`115200 8N1`). You may also mark them as reserved for SDK scripts.

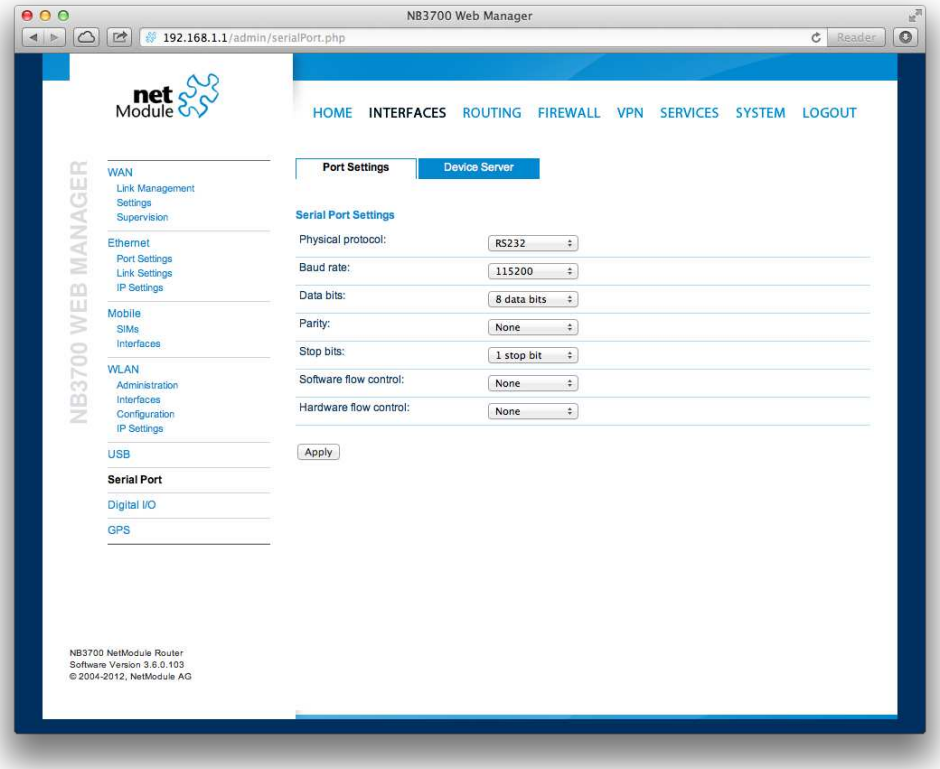


Figure 5.16.: Serial Port

Furtheron, a device server can be run for each port which can be used to control the serial device via IP.
It can be configured as follows:

Parameter	Serial Settings
Physical protocol	Selects the desired physical protocol on the serial port
Baud rate	Specifies the baud rate run on the serial port
Data bits	Specifies the number of data bits contained in each frame

Parameter	Serial Settings
Parity	Specifies the parity used for every frame that is transmitted or received
Stop bits	Specifies the number of stop bits used to indicate the end of a frame
Software flow control	Defines the software flow control for the serial port, `XOFF` will send a stop, `XON` a start character to the other end to control the rate of any incoming data
Hardware flow control	You may enable RTS/CTS hardware flow control, so that the RTS and CTS lines are used to control the flow of data
Protocol on TCP/IP	You may choose the IP protocols `Telnet` or `TCP raw` for the device server
Port	The TCP port for the device server
Timeout	The timeout until a client is declared as disconnected

The Serial Port is optional on NB3700.

5.3.7. Digital I/O

The Digital I/O page displays the current status of the I/O ports and can be used to turn output ports `on` or `off`.
You can apply the following settings:

Parameter	Digital I/O Settings
DO1 after reboot	Initial status of DO1 after system has booted
DO2 after reboot	Initial status of DO2 after system has booted

Besides `on` and `off` you may keep the `default` status as the hardware has initialized it after power-up.
The digital inputs and outputs can also be monitored and controlled by SDK scripts.

5.3.8. GNSS

Administration

The GPS page lets you enable or disable the GPS modules present in the system and can be used to configure the daemon that can be used to share access to receivers without contention or loss of data and to respond to queries with a format that is substantially easier to parse than the NMEA 0183 emitted directly by the GPS device.

We are currently running the Berlios GPS daemon (version 2.37), please navigate to `http://gpsd.berlios.de` for getting more information about how to incorporate it. The GPS values can also be queried by the CLI and used in SDK scripts.

Parameter	GPS Settings
Administrative status	Enable or disable GPS reception
Antenna type	The type of the connected GPS antenna, either active or passive
Server port	The TCP port on which the daemon is listening for incoming connections
Allow clients from	Specifies where clients can connect from, can be either `everywhere` or from a specific network
Clients start mode	Specifies how data transferal is accomplished when a client connects. You can specify `on request` which typically requires an `R` to be sent. Data will be sent instantly in case of `raw` mode which will provide NMEA frames or `super-raw` which includes the original data of the GPS receiver. If the client supports the JSON format (i.e. newer libgps is used) the `json mode` can be specified.

Information

This pages provides further information about the satellites in view and values derived from them:

Parameter	GPS Information
Latitude	The geographic coordinate specifying the north-south position
Longitude	The geographic coordinate specifying the east-west position

Parameter	GPS Information
Altitude	The height above sea level of the current location
Satellites in view	The number of satellites in view as stated in GPGSV frames
Speed	The horizontal and vertical speed in meter per second as stated in GPRMC frames
Satellites used	The number of satellites used for calculating the position as stated in GPGGA frames
Dilution of precision	The dilution of precision as stated in GPGSA frames

Furtheron, each satellite also comes with the following details:

Parameter	GPS Satellite Information
PRN	The PRN code of the satelitte (also referred as satellite ID) as stated in GPGSA frames
Elevation	The elevation (up-down angle between the dish pointing direction) in degrees as stated in GPGSV frames
Azimuth	The azimuth (rotation around the vertical axis) in degrees as stated in GPGSV frames
SNR	The SNR (Signal to Noise Ratio), often referred as signal strength

Please note that the values are shown as calculated by the daemon, their accuracy might be suggestive.

5.4. ROUTING

5.4.1. Static Routes

This menu shows all routing entries of the system. They are typically formed by an address/netmask couple (represented in IPv4 dotted decimal notation) which specify the destination of a packet. The packets can be directed to either a gateway or an interface or both. If interface is set to ANY, the system will choose the route interface automatically, depending on the best matching network configured for an interface.

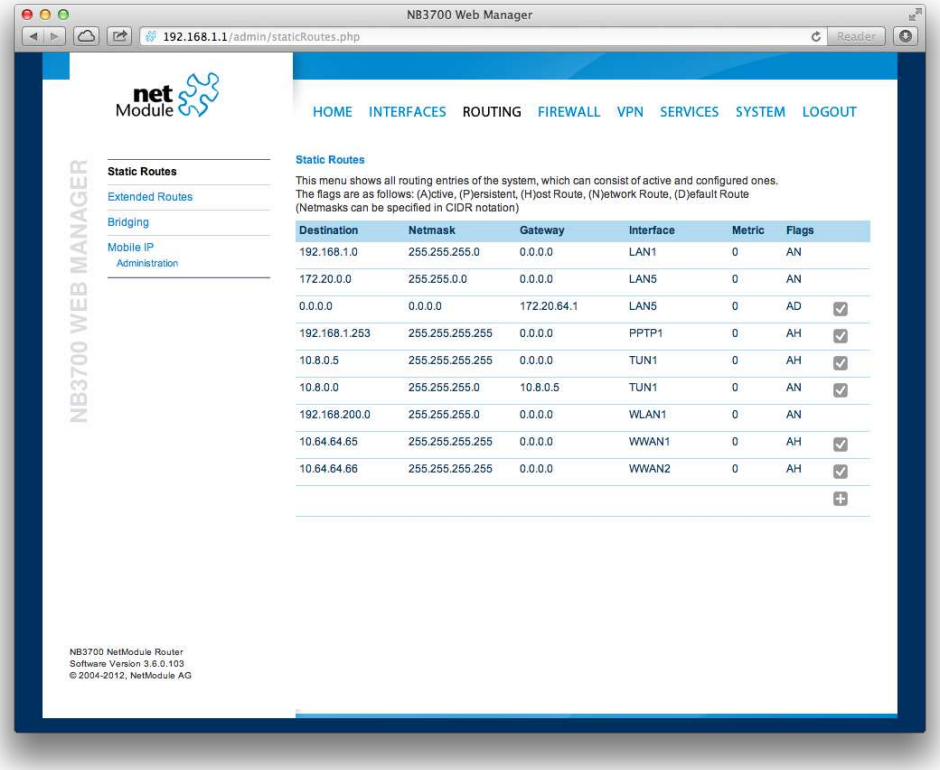

NB3700 Web Manager
192.168.1.1/admin/staticRoutes.php
Reader
net Module
HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM LOGOUT
NB3700 WEB MANAGER
Static Routes
Extended Routes
Bridging
Mobile IP
Administration
Static Routes
This menu shows all routing entries of the system, which can consist of active and configured ones.
The flags are as follows: (A)ctive, (P)ersistent, (H)ost Route, (N)etwork Route, (D)efault Route
(Netmasks can be specified in CIDR notation)
Destination Netmask Gateway Interface Metric Flags
192.168.1.0 255.255.255.0 0.0.0.0 LAN1 0 AN
172.20.0.0 255.255.0.0 0.0.0.0 LAN5 0 AN
0.0.0.0 0.0.0.0 172.20.64.1 LAN5 0 AD
192.168.1.253 255.255.255.255 0.0.0.0 PPTP1 0 AH
10.8.0.5 255.255.255.255 0.0.0.0 TUN1 0 AH
10.8.0.0 255.255.255.0 10.8.0.5 TUN1 0 AN
192.168.200.0 255.255.255.0 0.0.0.0 WLAN1 0 AN
10.64.64.65 255.255.255.255 0.0.0.0 WWAN1 0 AH
10.64.64.66 255.255.255.255 0.0.0.0 WWAN2 0 AH
NB3700 NetModule Router
Software Version 3.6.0.103
© 2004-2012, NetModule AG

Figure 5.17.: Static Routing

In general, host routes precede network routes and network routes precede default routes. Additionally, a metric can be used to determine the priority of a route, a packet will go in the direction with the lowest metric in case a destination matches multiple routes. Netmasks can be specified in CIDR notation (i.e. /24 expands to 255.255.255.0).

Parameter	Static Route Configuration
Destination	The destination address of a packet

Parameter	Static Route Configuration
Netmask	The subnet mask which forms, in combination with the destination, the network to be addressed. A single host can be specified by a netmask of 255.255.255.255, a default route corresponds to 0.0.0.0.
Gateway	The next hop which operates as gateway for this network (can be omitted on peer-to-peer links)
Interface	The network interface on which a packet will be transmitted in order to reach the gateway or network behind it
Metric	The routing metric of the interface (default 0), higher metrics have the effect of making a route less favorable
Flags	(A)ctive, (P)ersistent, (H)ost Route, (N)etwork Route, (D)efault Route

The flags obtain the following meanings:

Flag	Description
A	The route is considered active, it might be inactive if the interface for this route is not yet up.
P	The route is persistent, which means it is a configured route, otherwise it corresponds to an interface route.
H	The route is a host route, typically the netmask is set to 255.255.255.255.
N	The route is a network route, consisting of an address and netmask which forms the subnet to be addressed.
D	The route is a default route, address and netmask are set to 0.0.0.0, thus matching any packet.

Table 5.28.: Static Route Flags

5.4.2. Extended Routing

Extended routes can be used to perform policy-based routing, they generally precede static routes.

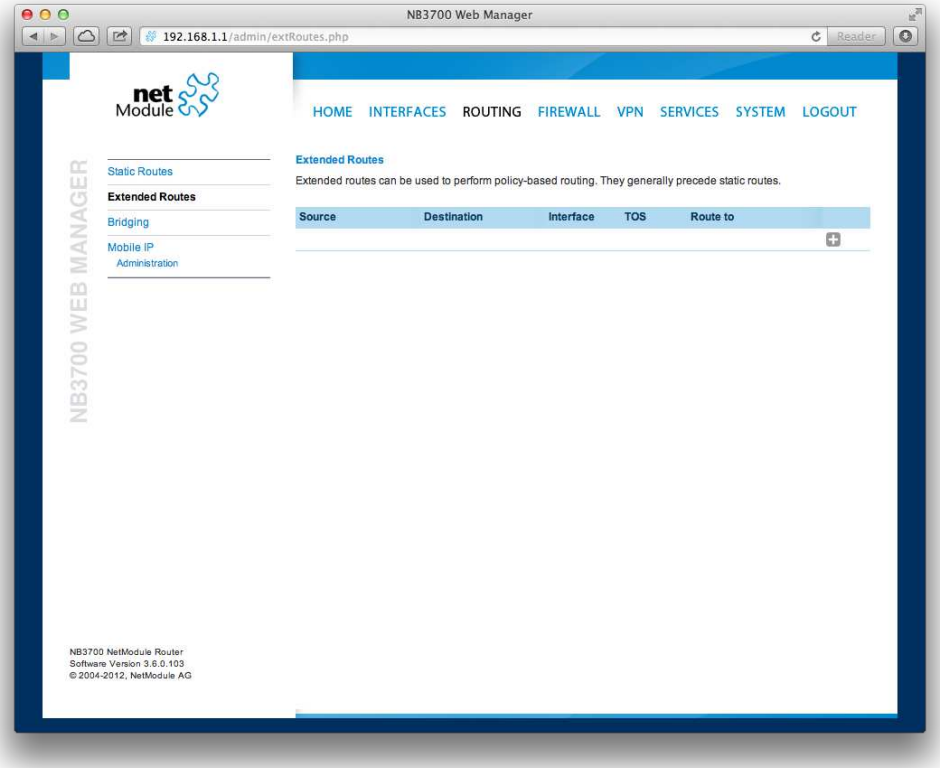


Figure 5.18.: Extended Routing

In contrast to statis routes, extended routes can be made up, not only of a destination address/netmask, but also a source address/netmask, incoming interface and the type of service (TOS) of packets.

Parameter	Extended Route Configuration
Source address	The source address of a packet
Source netmask	The source address of a packet
Destination address	The destination address of a packet
Destination netmask	The destination address of a packet
Incoming interface	The interface on which the packet enters the system
Type of service	The TOS value within the header of the packet

Parameter	Extended Route Configuration
Route to	Specifies the target interface or gateway to where the packet should get routed to

5.4.3. Multipath Routes

Multipath routes will perform weighted IP-session distribution for particular subnets across multiple interfaces.

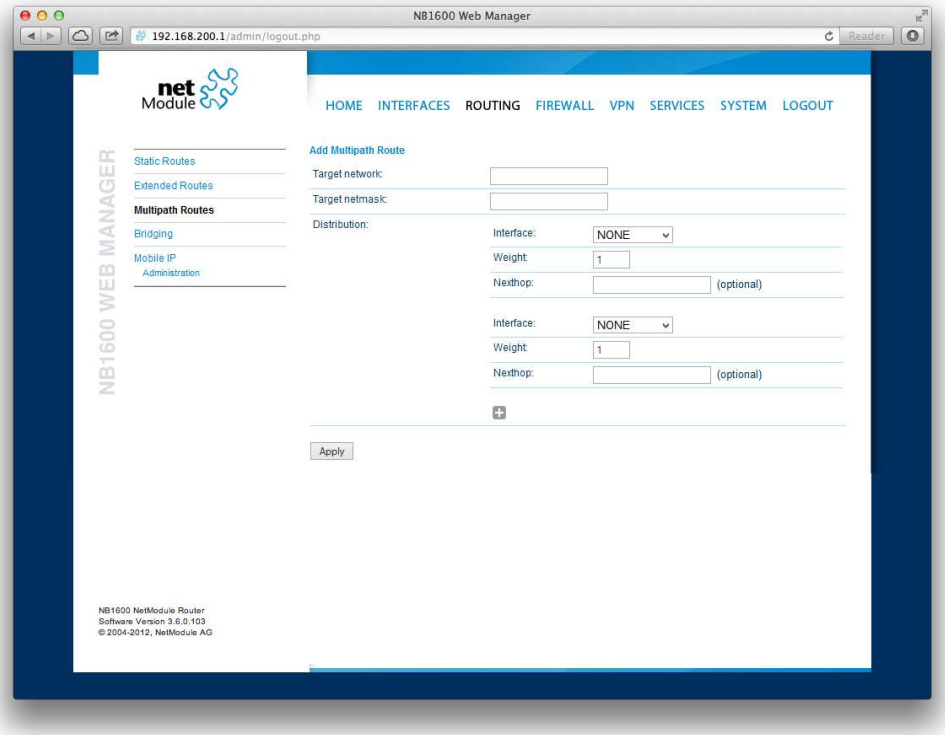


Figure 5.19.: Multipath Routes

At least two interfaces have to be defined to establish multipath routing. Additional interfaces can be added by pressing the plus sign.

Parameter	Add Multipath Routes
Target network/netmask	Defines the target network for which multipath routing shall be applied
Interface	Selects the interface for one path
Weight	Weight of the interface in relation to the others
NextHop	Overrides the default gateway of this interface

5.4.4. Mobile IP

Mobile IP (MIP) can be used to enable seamless switching between different kinds of WAN links (e.g. WWAN/WLAN). The `mobile node` hereby remains reachable via the same IP address (`home address`) at any time, independently of the WAN link being used. Effectively, any WAN link switch causes very small outages during switchover while keeping all IP connections alive.

Moreover, NetModule routers also support NAT-Traversal for mobile nodes running behind a firewall (performing NAT), which makes mobile nodes even there accessible from a central office via their home address, and thus, bypassing any complicated VPN setups.

The `home agent` accomplishes this by establishing a tunnel (similar to a VPN tunnel) between itself and the `mobile node`. WAN link switching works by telling the `home agent` that the WAN IP address (called the `care-of address` in MIP terms) of the `mobile node` has changed. The `home agent` will then encapsulate packets destined to a `mobile node`'s home address into a tunnel packet containing the current `care-of address` of the `mobile node` as its destination address.

To prevent problems with firewalls and private IP addressing, the MIP implementation always employs reverse tunneling, which means that all traffic sent by a `mobile node` is relayed via the tunnel to the `home agent` instead of directly being conveyed to the final destination. This fact also empowers MIP to be used as a lightweight VPN replacement (without payload secrecy).

The MIP implementation supports RFCs 3344, 5177, 3024 and 3519. For applications requiring vast numbers of mobile nodes, interoperability with the Cisco 2900 Series `home agent` implementation has been verified. However, since NetModule routers implement a `mobile node` as well as a `home agent`, a MIP network with up to 10 mobile nodes can be implemented without requiring expensive third party routers.

If MIP is run as a `mobile node`, the following settings can be configured:

Parameter	Mobile IP Configuration
Primary home agent address	The address of the primary `home agent`
Secondary home agent address	The address of the secondary `home agent`. The mobile node will try to register with this home agent, if the primary `home agent` is not reachable.
Home address	The permanent home address of the `mobile node` which can be used to reach the mobile router at any time

Parameter	Mobile IP Configuration
SPI	The Security Parameter Index (SPI) identifying the security context for the mobile IP tunnel between the `mobile node` and the `home agent`. This is used to distinguish mobile nodes from each other. Therefore each mobile node needs to be assigned a unique SPI. This is a 32-bit hexadecimal value.
Authentication type	The used authentication algorithm. This can be prefix-suffix-md5 (default for MIP) or hmac-md5.
Shared secret	The shared secret used for authentication of the `mobile node` at the `home agent`. This can be either a 128-bit hexadecimal value or a random length ASCII string.
Life time	The lifetime of security associations in seconds
UDP encapsulation	Specifies whether UDP encapsulation shall be used or not. To allow NAT traversal, UDP encapsulation must be enabled.
Mobile network address	Optionally specifies a subnet which should be routed to the `mobile node`. This information is forwarded via Network Mobility (NEMO) extensions to the `home agent`. The `home agent` can then automatically add IP routes to the subnet via the `mobile node`. Note that this feature is not supported by all third party `home agent` implementations.
Mobile network mask	The network mask for the optional routed network

If MIP is run as a `home agent`, you will have to set up a home address and network mask for the `home agent` first. Then you will need to add the configuration for all mobile nodes, which is made up of the following settings:

Parameter	Mobile IP Node Configuration
SPI	The Security Parameter Index (SPI) identifying the security context for the tunnel between the `mobile node` and the `home agent`. This is used to distinguish mobile nodes from each other. Therefore each `mobile node` needs to be assigned a unique SPI. This is a 32-bit hexadecimal value.

Parameter	Mobile IP Node Configuration
Authentication type	The used authentication algorithm. This can be prefix-suffix-md5 (default for mobile IP) or hmac-md5.
Shared secret	The shared secret used for authentication of the `mobile node` at the `home agent`. This can be either a 128-bit hexadecimal value or a random length ASCII string.

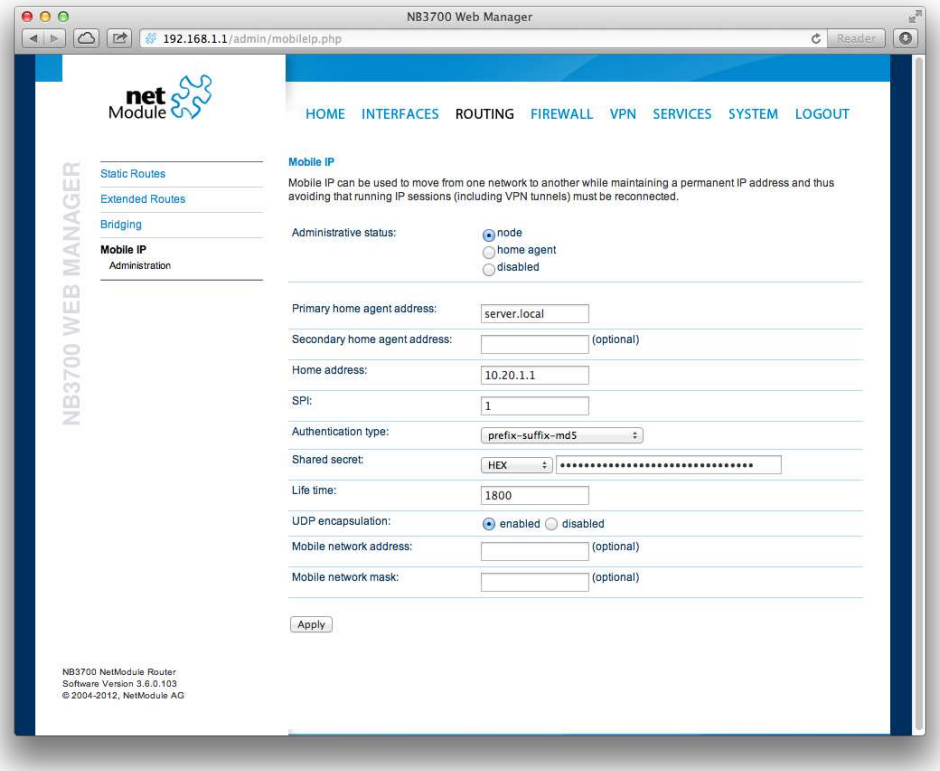


Figure 5.20.: Mobile IP

5.4.5. Quality Of Service

NetModule routers are able to prioritize and shape certain kinds of IP traffic. This is currently limited on egress, which means that only outgoing traffic can be stipulated.

The current QoS implementation uses Stochastic Fairness Queueing (SFQ) classes in combination with Hierarchy Token Bucket (HTB) queuing disciplines. In case of demands for other classes or qdiscs please contact our support team in order to evaluate the best approach for your application.

QoS Administration

The administration page can be used to enable and disable QoS.

QoS Classification

The classification section can be used to define the WAN interfaces on which QoS should be active.

Parameter	QoS Interface Parameters
Interface	The WAN interface on which QoS should be active
Bandwidth congestion	The bandwidth congestion method. In case of `auto` the system will try to apply limits in a best-effort way. However, it is suggested to set fixed bandwidth limits as they also offer a way of tuning the QoS behaviour.
Downstream bandwidth	The available bandwidth for incoming traffic
Upstream bandwidth	The available bandwidth for outgoing traffic

When defining limits, you should consider bandwidth limits which are at least possible as most shaping and queues algorithms will not work correctly if the specified limits cannot be achieved. In particular, any WWAN interfaces operating in a mobile enviroment are suffering variable bandwidths, thus rather lower values should be used.

In case an interface has been activated, the system will automatically create the following queues:

Parameter	QoS Default Queues
high	A high priority queue which may hold any latency-critical services (such as VoIP)
default	A default queue which will handle all other services
low	A low priority queue which may hold less-critical services for which shaping is intended

Each queue can be configured as follows:

Parameter	QoS Queue Parameters
Name	The name of the QoS queue
Priority	A numerical priority for the queue, lower values indicate higher priorities
Bandwidth	The maximum possible bandwidth for this queue

You can now configure and assign any services to each queue. The following parameters apply:

Parameter	QoS Service Parameters
Interface	The QoS interface of the queue
Queue	The QoS queue to which this service shall be assigned
Source	Specifies a network address and netmask used to match the source address of packets
Destination	Specifies a network address and netmask used to match the destination (target) address of packets
Protocol	Specifies the protocol for packets to be matched
Type of Service	Specifies the TOS/DiffServ for packets to be matched

5.5. FIREWALL

5.5.1. Administration

NetModule routers use Linux's netfilter/iptables firewall framework (see http://www.netfilter.org for more information) which supports stateful inspection, that is, granting the same permissions for inherited connections within an IP session (e.g. FTP which builds up a control and data connection).
The administration page can be used to enable and disable firewalling. When turning it on, a shortcut can be used to generate a predefined set of rules which allow administration (over HTTP, HTTPS, SSH or TELNET) by default but block any other packets coming from the WAN interface.

5.5.2. Adress Groups

This menu can be used to form address groups which can be later used for firewall rules in order to reduce the number of rules for a set of addresses.

5.5.3. Rules

In general, the firewall is set up of a range of rules which control each packet's permission to pass the router. Please note that the rules are processed by order, that means traversing the list from top to bottom until a matching rule is found. Packets which are not matching any of the rules configured will be ALLOWED.

Parameter	Firewall Rule Configuration
Description	A meaningful description about the purpose of this rule
Mode	Specifies whether the packets of this rule should be allowed or denied
Source	The source address of matching packets, can be any or specified by address/network
Destination	The destination address of matching packets, can be any, local (addressed to the system itself) or specified by address/network
Incoming interface	The interface on which matching packets are received
Protocol	The used IP protocol of matching packets (UDP, TCP or ICMP)

Parameter	Firewall Rule Configuration
Destination port(s)	The destination port of matching packets, which can be specified by a single port or a range of ports (only UDP/TCP)

The statistics page can be used to figure out if rules have matched any packets and provides a convenient way to debug your firewall setup.

5.5.4. NAPT

This page can be used to configure Network Address and Port Translation (NAPT) for packets traversing the system. NAPT hereby modifies IP addresses or/and TCP/UDP ports in matching IP packets. By tracking those connections, it will also automatically adjust the returning packets of an IP session.

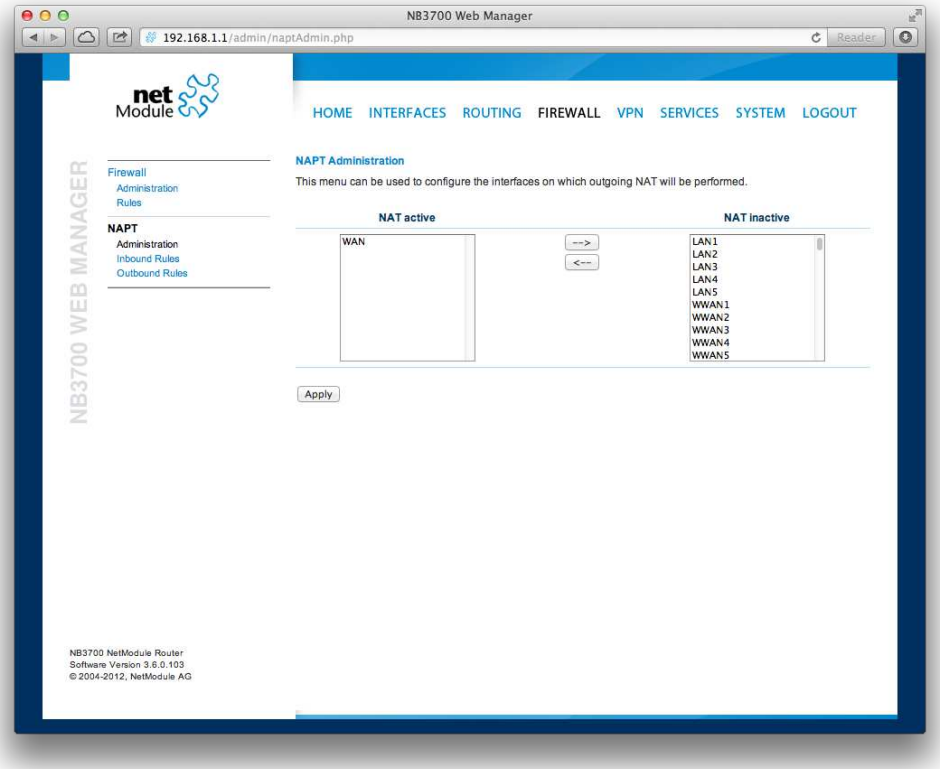


Figure 5.21.: NAPT Administration

The administration page lets you specify the interfaces on which outgoing NAT (also called *Masquerading*) will be performed. NAT will hereby use the address of the selected interface and choose a random source port for outgoing connections and thus enables communication between hosts from a private local area network towards hosts on the public network.

NAPT Inbound Rules

Inbound rules can be used to modify the target section of IP packets and, for instance, forward a service or port to an internal host. By doing so, you can expose that service and make it available from the Internet. You may also establish 1:1 NAT mapping for a single host using additional outbound rules.

Please note that the specified rules are processed by order, that means, traversing the

list from top to bottom until a matching rule is found. If there is no matching rule found, the packet will pass as is.

Parameter	Inbound NAPT Rules
Description	A meaningful description of this rule
Incoming interface	The interface from which matching packets are received
Target address	The destination address of matching packets (optional)
Protocol	The used protocol of matching packets
Ports	The used UDP/TCP port of matching packets
Redirect to	The address to which matching packets shall be redirected
Redirect port	The port to which matching packets will be redirected

NAPT Outbound Rules

Outbound rules will modify the source section of IP packets and can be used to establish 1:1 NAT mappings but also to redirect packets to a specific service.

Parameter	Outbound NAPT Rules
Description	A meaningful description of this rule
Incoming interface	The outgoing interface on which matching packets are leaving the router
Source address	The source address of matching packets (optional)
Protocol	The used protocol of matching packets
Ports	The used UDP/TCP port of matching packets
Rewrite source address	The address to which the source address of matching packets shall be rewritten
Rewrite source port	The port to which the source port of matching packets shall be rewritten

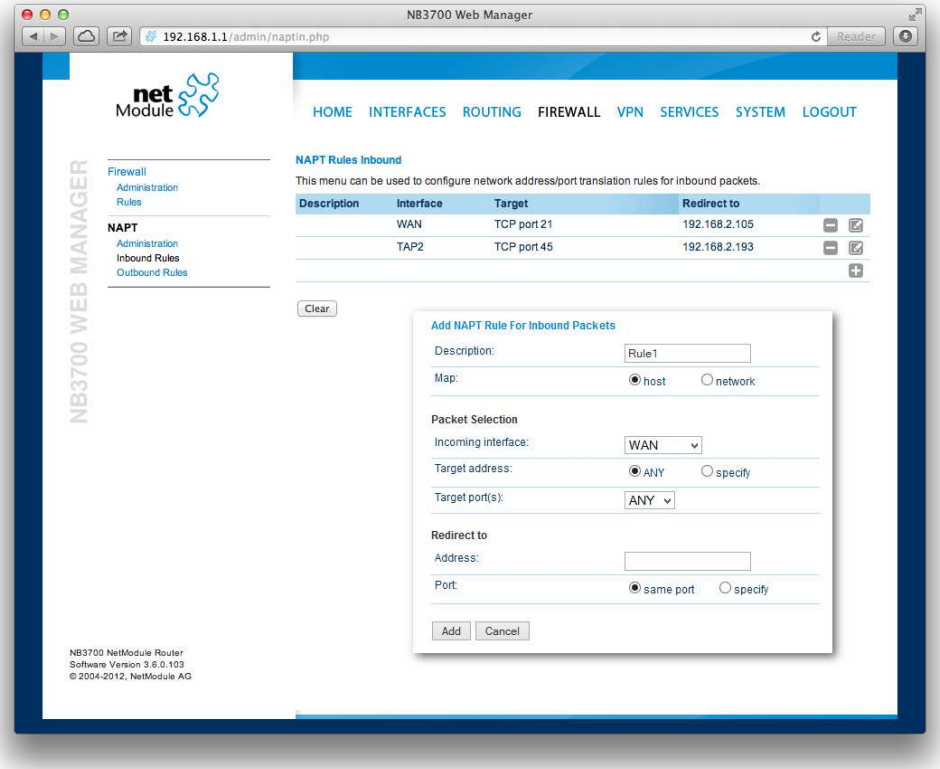


Figure 5.22.: Inbound NAPT

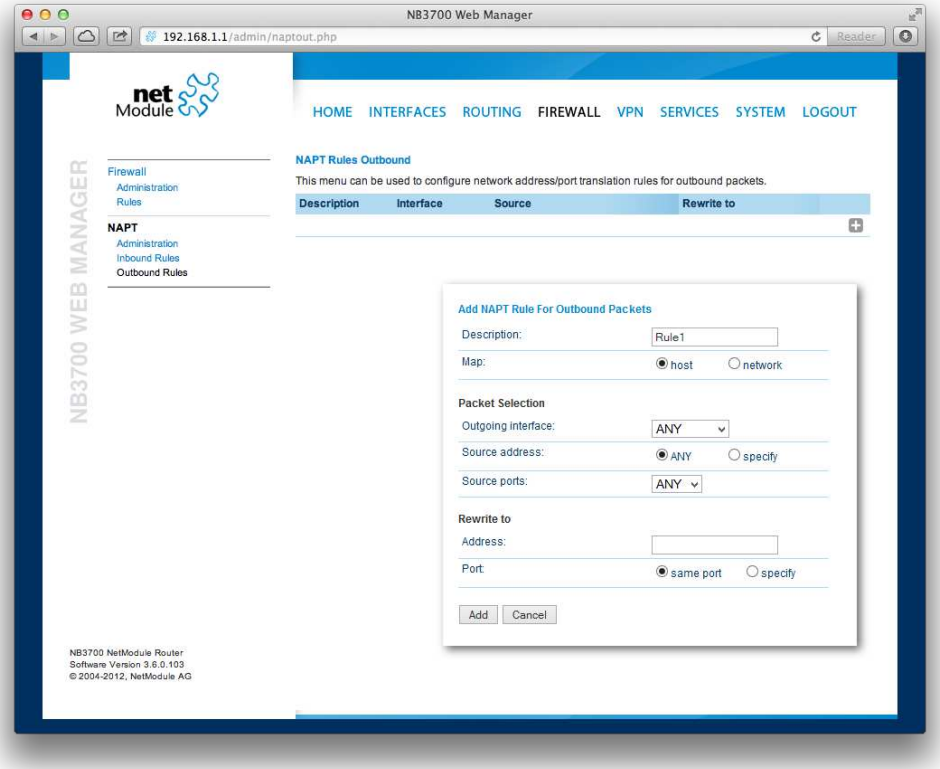


Figure 5.23.: Outbound NAPT

5.6. VPN

5.6.1. OpenVPN

OpenVPN Administration

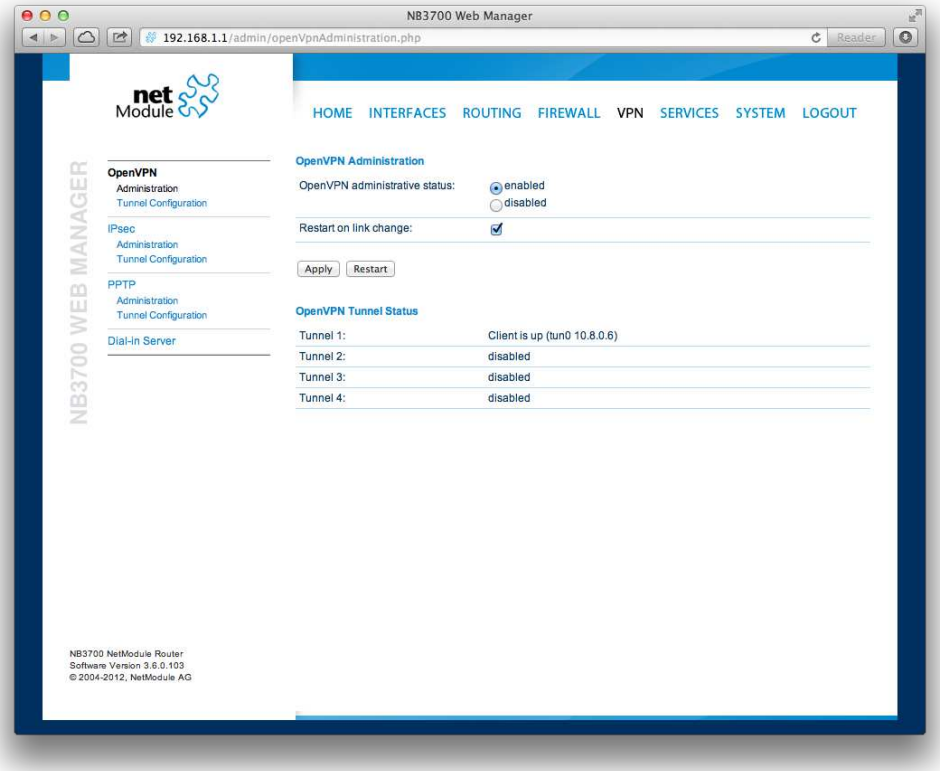


Figure 5.24.: OpenVPN Administration

Tunnel Configuration

NetModule routers support one single server tunnel and up to four client tunnels. You can specify tunnel parameters either in standard configuration or upload an expert mode file which has been created in advance. Refer to chapter 5.6.1 to learn more about how to manage clients and generate the files.

Parameter	OpenVPN Configuration
Operation mode	Specifies whether client or server mode should be used for this tunnel, it further specifies if tunnel shall be configured in a standard way or if an expert mode file shall be used.

If the tunnel is operated in client mode, the following settings can be applied:

Parameter	OpenVPN Client Configuration
Peer selection	Specifies how the remote peer shall be selected, besides a single server you may configure multiple servers which can, in case of failures, either be selected sequently (i.e. failover) or randomly (i.e. load balancing)
Server	The address or hostname of the remote server
Port	The port of the remote server (1194 by default)

Setting up a tunnel server just requires the server port to be set, the settings mentioned below apply for both, server and client tunnels:

Parameter	OpenVPN Configuration
Type	The device type for this tunnel which can be either TUN (typically used for routed connections) or TAP (required for bridged networks)
Protocol	The tunnel protocol to be used for the transport connection
Network mode	Defines how the packets should be forwarded, which can be either routed or bridged from/to a particular LAN interface. If required, you can also specify the maximum transfer unit for the tunnel interface.
Authentication	You can choose between credential-based (where you have to specify a username and password) but we generally suggest to use certificate-based authentication. Note that keys/certificates have to be created or uploaded (see 5.8.6). You may also specify which message digest shall be used for authenticating packets.
Cipher	The required cipher mechanism used for encryption
Use compression	Enable or disable packet compression
Use keepalive	Can be used to send a periodic keepalive packet in order to keep the tunnel up despite of inactivity

Parameter	OpenVPN Configuration
Redirect gateway	By redirecting the gateway, all packets will be directed to the VPN tunnel. Please ensure that essential services (such as DNS or NTP servers) can be reached at the network behind the tunnel. In doubt, create an extra static route pointing to the correct interface.

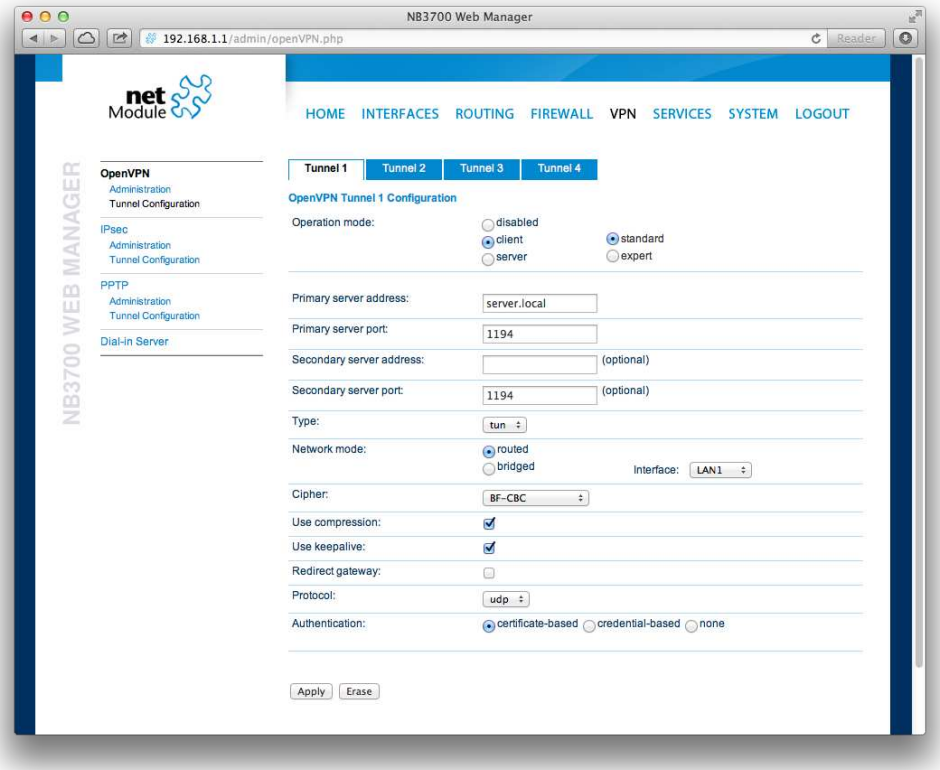


Figure 5.25.: OpenVPN Configuration

ExpertConfiguration

OpenVPN Expert Configuration (Client)

The expert configuration mode offers a straightforward way to configure a tunnel by simply uploading a package containing the required configuration and optionally key/certificate files. A client tunnel usually consists of the following files:

Parameter	Client Expert Files
client.conf	OpenVPN configuration file (see http://www.openvpn.net for available options)
ca.crt	aoot certificate authority file
client.crt	Certificate file
client.key	Private key file

Please note that you may specify arbitrary file names, however, the configuration file suffix must be `.conf` and all files referred in the configuration file must correspond to relative path names.

OpenVPN Expert Configuration (Server) A server tunnel typically requires the following files:

Parameter	erver Expert Files
server.conf	OpenVPN configuration file
ca.crt	Root certificate authority file
server.crt	Certificate file
server.key	Private key file
dh1024.pem	Diffie-Hellman parameters file
ccd	A directory containing client-specific configuration files

Keep in mind that a certificate becomes valid once its validity time has been reached, thus an accurate system has to be set prior to creating certificates and establishing a tunnel connection. Please ensure that all NTP servers are reachable. Using host names also requires a working DNS server.

Client Management

Once you have successfully set up an OpenVPN server tunnel, you can manage and enable clients connecting to your service. Currently connected clients can be seen on this page, including the connect time and IP address. You may kick connected clients by disabling them.

In the Networking section you can specify a fixed tunnel endpoint address for each client. Please note that, if you intend to use a fixed address for a particular client, you would have to apply fixed addresses to the other ones as well.

You may specify the network behind the clients as well as the routes to be pushed to each

client. This can be useful for routing purposes, e.g. in case you want to redirect traffic for particular networks towards the server. Routing between the clients is generally not allowed but you can enable it if desired.

Finally, you can generate and download all expert mode files for enabled clients which can be used to easily populate each client.

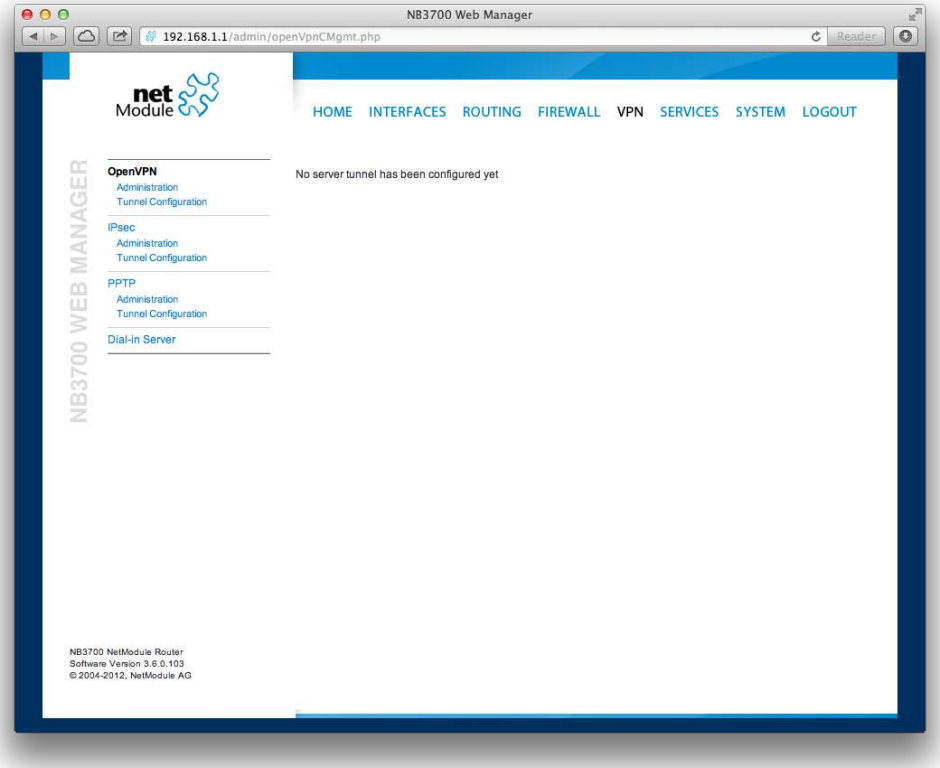


Figure 5.26.: OpenVPN Client Management

5.6.2. IPsec

IPsec is a protocol suite for securing IP communications by authenticating and encrypting each packet of a communication session and thus establishing a secure virtual private network.
IPsec includes various cryptographic protocols and ciphers for key exchange and data encryption and can be seen as one of the strongest VPN technologies in terms of security. It uses the following mechanisms:

Mechanism	Description
AH	Authentication Headers (AH) provide connectionless integrity and data origin authentication for IP datagrams and ensure protection against replay attacks.
ESP	Encapsulating Security Payloads (ESP) provide confidentiality, data-origin authentication, connectionless integrity, an anti-replay service and limited traffic-flow confidentiality.
SA	Security Associations (SA) provide a secure channel and a bundle of algorithms that provide the parameters necessary to operate the AH and/or ESP operations. The Internet Security Association Key Management Protocol (ISAKMP) provides a framework for authenticated key exchange.

Negotating keys for encryption and authentication is generally done by the Internet Key Exchange protocol (IKE) which consists of two phases:

Phase	Description
IKE phase 1	IKE authenticates the peer during this phase for setting up an ISAKMP secure association. This can be carried out by either using `main` or `aggressive` mode. The `main` mode approach utilizes the Diffie-Hellman key exchange and authentication is always encrypted with the negotiated key.The `aggressive` mode just uses hashes of the pre-shared key and therefore represents a less-secure mechanism which should generally be avoided as it is prone to dictionary attacks.
IKE phase 2	IKE finally negotiates IPSec SA parameters and keys and sets up matching IPSec SAs in the peers which is required for AH/ESP later on.

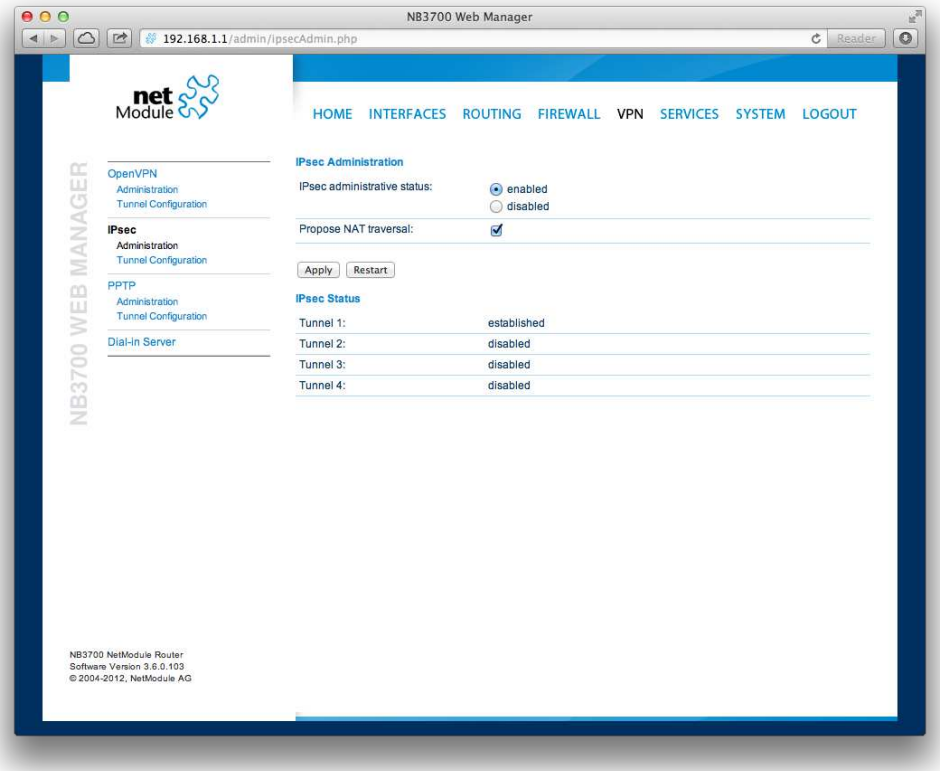


Figure 5.27.: IPsec Administration

Administration

This page can be used to enable/disable IPsec, you may also specify whether NAT-Traversal should be used.

NAT-Traversal is mainly used for connections which traverse a path where a router modifies the IP address/port of packets. It encapsulates packets in UDP and therefore requires a slight overhead which has to be taken into account when running over small-sized MTU interfaces.

Please note that running NAT-Traversal makes IKE using UDP port 4500 rather than 500 which has to be taken into account when setting up firewall rules.

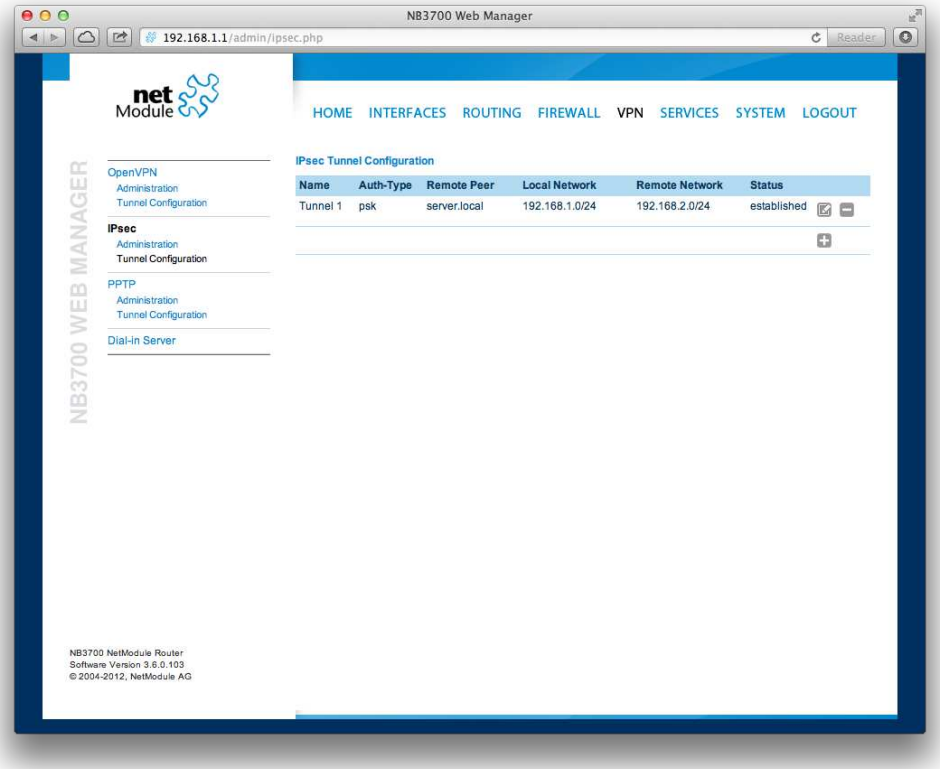


Figure 5.28.: IPsec Configuration

General

For setting up the tunnel you will have to configure the following parameters first:

Parameter	IPsec General Settings
Remote peer	IP address or host name of the remote IPsec peer. You may specify 0.0.0.0 to act as a responder for roadwarrior clients.

Parameter	IPsec General Settings
DPD Status	Specifies whether Dead Peer Detection (see RFC 3706) shall be used. DPD will detect any broken IPSec connections, in particular the ISAKMP tunnel, and refresh the corresponding SAs (Security Associations) and SPIs (Security Payload Identifier) for a faster re-establishment of the tunnel.
Detection cycle)	The delay (in seconds) between DPD keepalives that are sent for this connection (default 30 seconds)
Failure threshold	The number of unanswered DPD requests until the IPsec peer is considered dead (the router will then try to re-establish a dead connection automatically)

IKE Authentication

NetModule routers support IKE authentication through pre-shared keys (PSK) or certificates within a public key infrastructure.

Using PSK requires the following settings:

Parameter	IPsec IKE Authentication Settings
PSK	The pre-shared key used to authenticate at the peer
Local ID Type	The type of identification for the local ID which can be a `FQDN`, `username@FQDN` or `IP address`
Local ID	The local ID value
Local ID Type	The type of identification for the remote ID
Remote ID	The remote ID value

When using certificates you would need to specify the operation mode. When run as PKI client you can create a Certificate Signing Request (CSR) in the certificates section which needs to be submitted at your Certificate Authority and imported to the router afterwards. In PKI server mode the router represents the Certificate Authority and issues the certificates for remote peers.

IKE Proposal

This section can be used to configure the phase 1 settings:

Parameter	IPsec IKE Proposal Settings
Negotiation mode	Choose the desired negotiation mode. Preferably, `main` mode should be used but `aggressive` mode might be applicable when dealing with dynamic endpoint addresses.
Encryption algorithm	The desired IKE encryption method (we recommend AES256)
Authentication algorithm	The desired IKE authentication method (we prefer SHA1 over MD5)
IKE Diffie-Hellman Group	The IKE Diffie-Hellman Group
SA life time	The lifetime of Security Associations
Perfect Forward Secrecy	Specifies whether Perfect Forward Secrecy (PFS) should be used. This feature increases security as PFS avoids penetration of the key-exchange protocol and prevents compromisation of previous keys.

IPsec Proposal

This section can be used to configure the phase 2 settings:

Parameter	IPsec Proposal Settings
Encapsulation mode	The desired encapsulation mode (Tunnel or Transport)
IPsec protocol	The desired IPsec protocol (AH or ESP)
Encryption algorithm	The desired IKE encryption method (we recommend AES256)
Authentication algorithm	The desired IKE authentication method (we prefer SHA1 over MD5)
SA life time	The lifetime of Security Associations

Networks

When creating Security Associations, IPsec will keep track of routed networks within the tunnel. Packets will be only transmitted when a valid SA with matching source and destination network is present. Therefore, you may need to specify the networks right and left of the endpoints by applying the following settings:

Parameter	IPsec Network Settings
Local network address	The address of your local area network
Local network mask	The netmask of your local area network
Peer network address	The address of the remote network behind the peer
Peer network mask	The netmask of the remote network behind the peer
NAT address	Optionally, you can apply NAT (masquerading) for packets coming from a different local network. The NAT address must reside in the network previously specified as local network.

5.6.3. PPTP

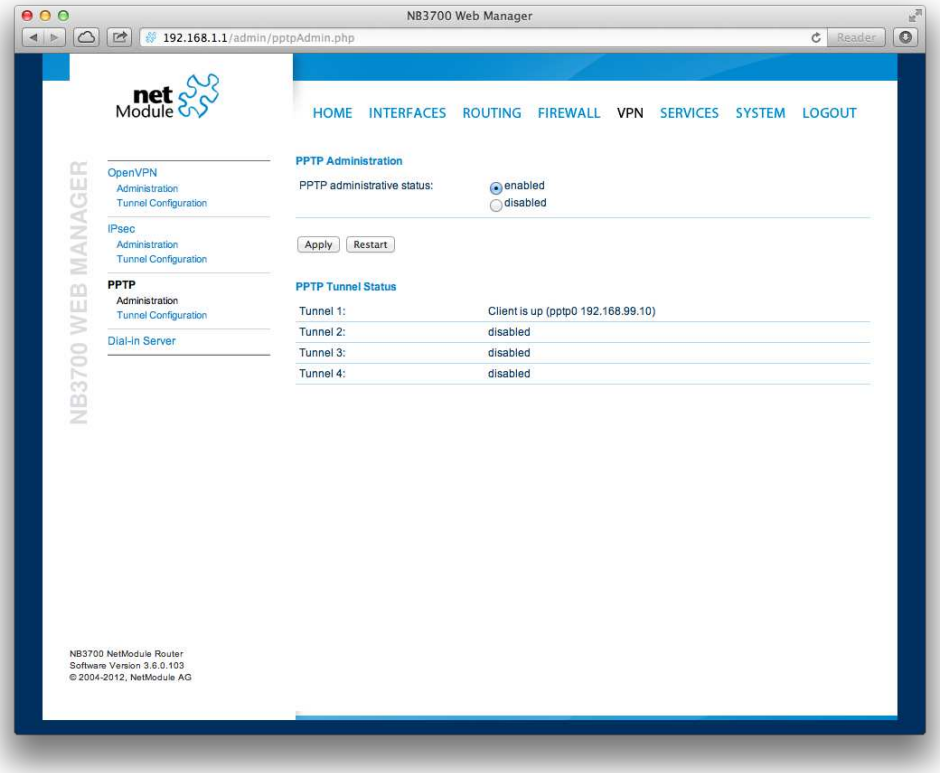


Figure 5.29.: PPTP Administration

The Point-to-Point Tunneling Protocol (PPTP) is a method for implementing virtual private networks between two hosts. PPTP is easy to configure and widely deployed amongst Microsoft Dial-up networking servers. However, due to its weak encryption algorithms, it is nowadays considered insecure but it still provides a straightforward way for establishing tunnels.

When setting up a PPTP tunnel, you would need to choose between server or client. A client tunnel requires the following parameters to be set:

Parameter	PPTP Client Settings
Server address	The address of the remote server
Username	The user-name used for authentication
Password	The password used for authentication

Setting up a server requires the following settings:

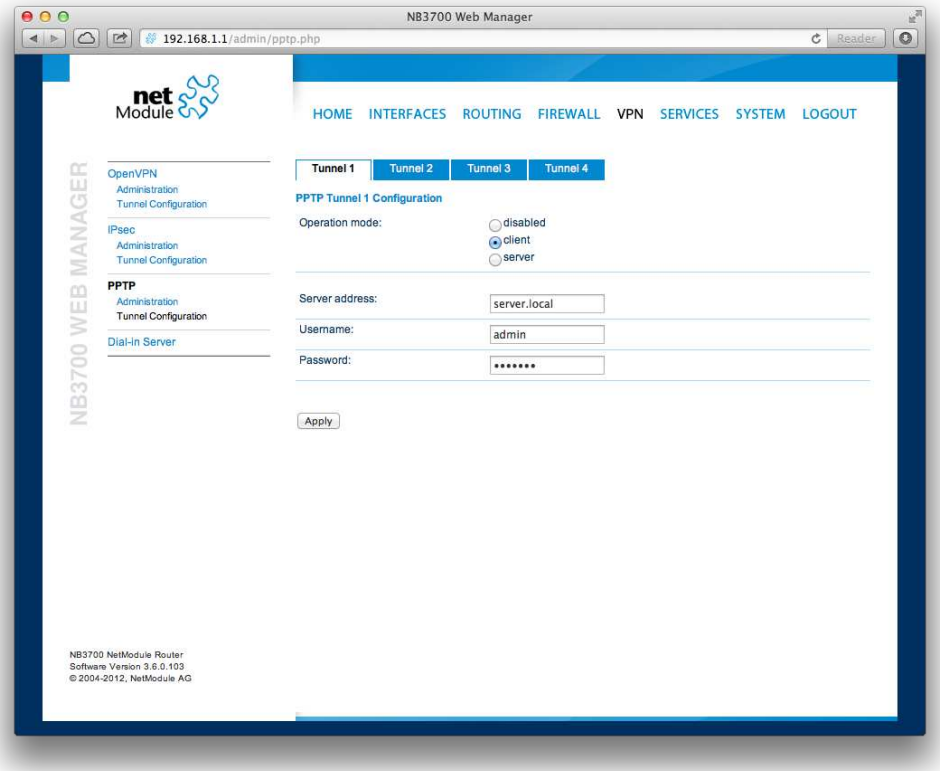


Figure 5.30.: PPTP Tunnel Configuration

Parameter	PPTP Server Settings
Listen address	Specifies on which IP address should be listened for incoming client connections
Server address	The server address within the tunnel
Client address range	Specifies a range of IP addresses assigned to each client

PPTP Client Management

PPTP clients for a server tunnel need to be configured here. They are made up of username and password. A fixed IP address can be assigned to them which can be used to point any routes to a dedicated tunnel.

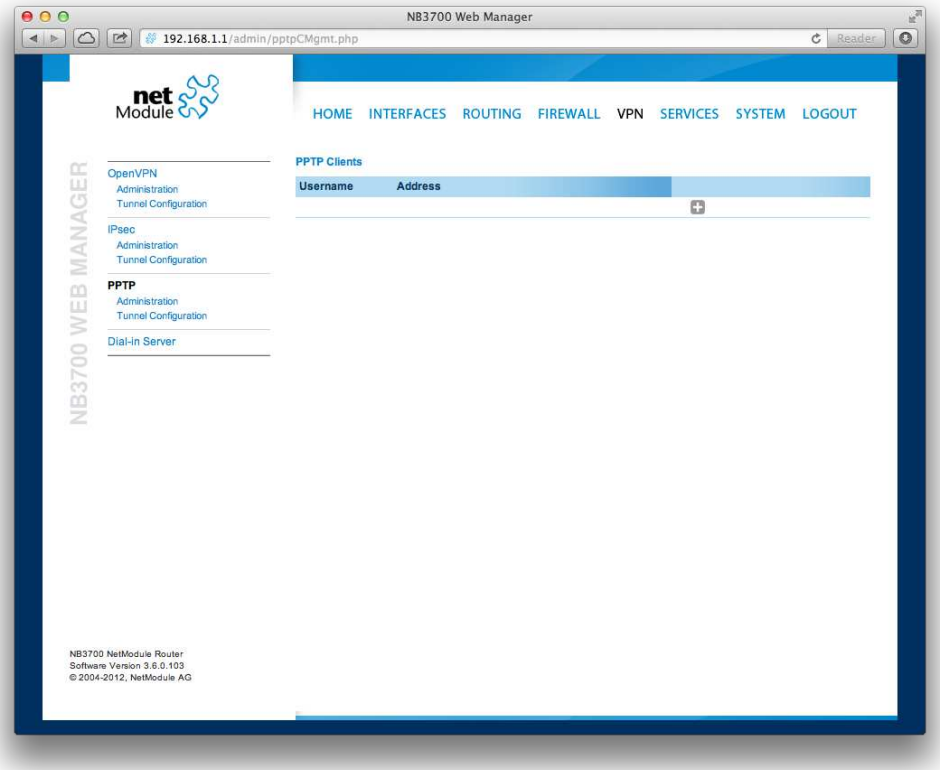


Figure 5.31.: PPTP Client Management

5.6.4. GRE

The Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over IP. GRE is defined in RFC 1701, 1702 and 2784. It does not provide encryption nor authorization but can be used on an address-basis on top of other VPN techniques (such as IPsec) for tunneling purposes.

The following parameters are required for setting up a tunnel:

Parameter	GRE Configuration
Peer address	The IP address of the remote peer
Local tunnel address	The local IP address of the tunnel
Local tunnel netmask	The local subnet mask of the tunnel
Remote network	The remote network address of the tunnel
Remote netmask	The remote subnet mask of the tunnel

In general, the local tunnel address/netmask should not conflict with any other interface addresses. The remote network/netmask will result in an additional route entry in order to control which packets should be encapsulated and transferred over the tunnel.

5.6.5. Dial-In

On this page you can configure the Dial-In server in order to establish a data connection over GSM calls. Thus, one would generally apply a required service type of 2G-only, so that the modem registers to GSM only. Naturally, a concurrent use of outgoing WWAN interfaces and Dial-In connection is not possible.

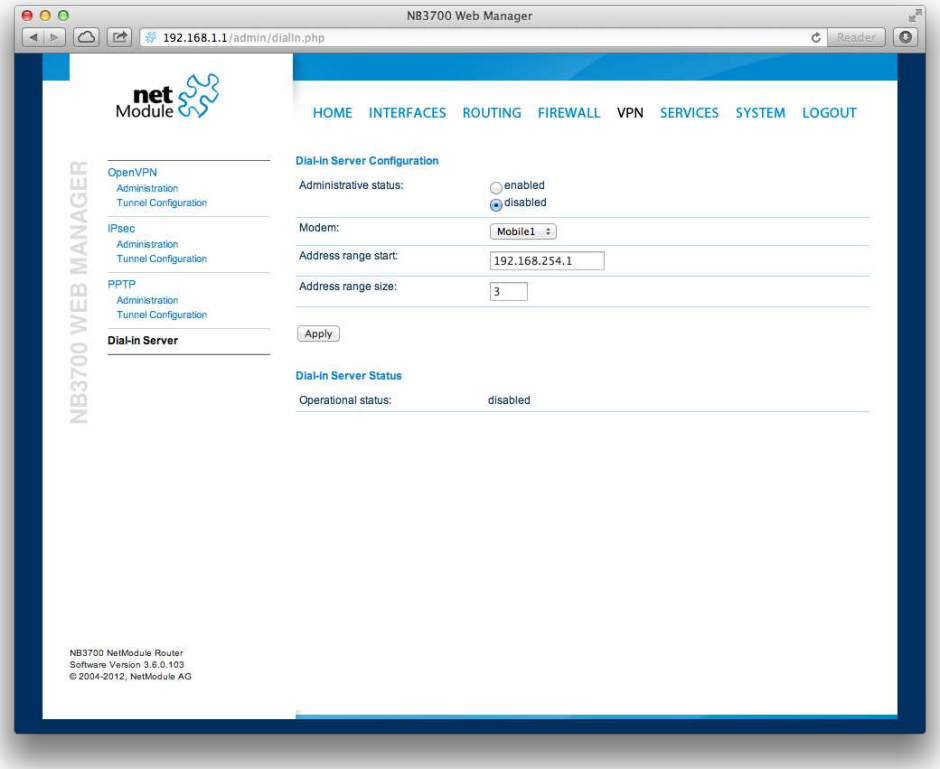


Figure 5.32.: Dial-in Server Settings

The following settings can be set:

Parameter	Dial-in Server Configuration
Administrative status	Specifies whether incoming calls shall be answered or not
Modem	Specifies the modem on which calls can come in
Address range start	Start of the IP address range assigned to incoming clients
Address range size	Number of addresses for client IP address range

Besides the admin account you can configure further users in the user accounts section which shall be allowed to dial-in.

Please note that Dial-In connections are generally discouraged. As they are implemented as GSM voice calls, they suffer from unreliability and poor bandwidth.

5.7. SERVICES

5.7.1. SDK

NetModule routers are shipping with a Software Development Kit (SDK) which offers a simple and fast way to implement customer-specific functions and applications. It consists of:

1. An SDK host which defines the runtime environment (a so-called sandbox), that is, controlling access to system resources (such as memory, storage and CPU) and, by doing so, catering for the right scalability
2. An interpreter language called `arena`, a light-weight scripting language optimized for embedded systems, which uses a syntax similar to ANSI-C but adds support for exceptions, automatic memory management and runtime polymorphism on top of that
3. A NetModule-specific Application Programming Interface (API), which ships with a comprehensive set of functions for accessing hardware interfaces (e.g. digital IO ports, GPS, external storage media, serial ports) but also for retrieving system status parameters, sending E-Mail or SMS messages or simply just to configure the router

Anyone, reasonably experienced in the C language, will find an environment that is easy to dig in. However, feel free to contact us via router@support.netmodule.com and we will happily support you in finding a programming solution to your specific problem.

The Language

The `arena` scripting language offers a broad range of POSIX functions (like `printf` or `open`) and provides, together with tailor-made API functions, a simple platform for implementing any sort of applications to interconnect your favourite device or service with the router.

Here comes a short example:

```
/* We are going to eavesdrop on the first serial port
 * and turn on lights via a digital I/O output port,
 * otherwise we'd have to send a short message.
 */

for (attempts = 0; attempts < 3; attempts++) {
  if (nb_serial_read("serial0") == "Knock Knock!") {
    nb_serial_write("serial0", "Who's there?");

    if (nb_serial_read("serial0") == "Santa") {
      printf("Hurray!\n");
      nb_dio_set("out1", 1);
    }
  }
}
nb_sms_send("+123456789", "No presents this year :(")
```

A set of example scripts can be downloaded directly from the router, you can find a list of them in the appendix. The manual which can be obtained from the NetModule support web page gives a detailed introduction of the language, including a description of all available functions.

SDK API Functions

The current range of API functions can be used to implement the following features:

1. Send/Retrieve SMS
2. Send E-mail
3. Read/Write from/to serial device
4. Control digital input/output ports
5. Run TCP/UDP servers
6. Run IP/TCP/UDP clients
7. Access files of mounted media (e.g. an USB stick)
8. Retrieve status information from the system
9. Get or set configuration parameters
10. Write to syslog
11. Transfer files over HTTP/FTP
12. Get system events / Reboot system
13. Control the LEDs

The SDK API which can be obtained from the NetModule support page provides an overview but also explains all functions in detail.

Please note that some functions require the corresponding services (e.g. E-Mail, SMS) to be properly configured prior to utilizing them in the SDK.

Let's now pay some attention to the very powerful API function `nb_status`. It can be used to query the router's status values in the same manner as they can be shown with the CLI. It returns a structure of variables for a specific section (a list of available sections can be obtained by running `cli status -h`).

By using the `dump` function you can figure out the content of the returned structure:

```
/* dump current location */

dump(nb_status("location"));
```

The script will then generate lines like maybe these:

```
struct(8): {
  .LOCATION_STREET       = string[11]: "Bahnhofquai"
  .LOCATION_CITY         = string[10]: "Zurich"
  .LOCATION_COUNTRY_CODE = string[2]:  "ch"
  .LOCATION_COUNTRY      = string[11]: "Switzerland"
  .LOCATION_POSTCODE     = string[4]:  "8001"
  .LOCATION_STATE        = string[6]:  "Zurich"
  .LOCATION_LATITUDE     = string[9]:  "47.3778058"
  .LOCATION_LONGITUDE    = string[8]:  "8.5412757"
}
```

In combination with the `nb_config_set` function, it is possible to start a re-configuration of any parts of the system upon status changes. You may query possible sections and parameters again with the CLI:

```
~ $ cli get -c wanlink.0
Showing configuration sections (matching 'wanlink.0'):

wanlink.0.mode
wanlink.0.name
wanlink.0.prio
wanlink.0.weight
```

Running the CLI in interactive mode, you will be also able to step through possible configuration parameters by the help of the `TAB` key.

Here is an example how one might adopt those functions:

```
/* check current city and enable the second WAN link */

location = nb_status("location");
if (location) {
    city = struct_get(location, "LOCATION_CITY");

    if (city == "Wonderland") {
        for (led = 0; led < 5; led++) {
            nb_led_set(led, LED_BLINK_FAST|LED_COLOR_RED);
        }
    } else {
        printf("You'll never walk alone in %s ...\n", city);
        nb_config_set("wanlink.1.mode=1");
    }
}
```

Running SDK

In the SDK, we are speaking of `scripts` and `triggers` which form `jobs`.

Any `arena` script can be uploaded to the router or imported by using dedicated user configuration packages. You may also edit the script directly at the Web Manager or select one of our examples. You will further have a testing section on the router which can be used to check your syntax or doing test runs.

Once uploaded, you will have to specify a trigger, that is, telling the router when the script is to be executed. This can be either time-based (e.g. each Monday) or triggered by one of the pre-defined system events (e.g. wan-up) as described in Events chapter 5.7.7. With both, a script and a trigger, you can finally set up an SDK job now. The `test` event usually serves as a good facility to check whether your job is doing well. The admin section also offers facilities to troubleshoot any issues and control running jobs.

The SDK host (sdkhost) corresponds to the daemon managing the scripts and their operations and thus avoiding any harm to the system. In terms of resources, it will limit CPU and memory for running scripts and also provide a pre-defined portion of the available flash storage. You may, however, extend it by external USB storage or (depending on your model) SD cards.

Files written to `/tmp` will be hold in memory and will be cleared upon a restart of the script. As your scripts operate in the sandbox, you will have no access to tools on the system (such as `ifconfig`).

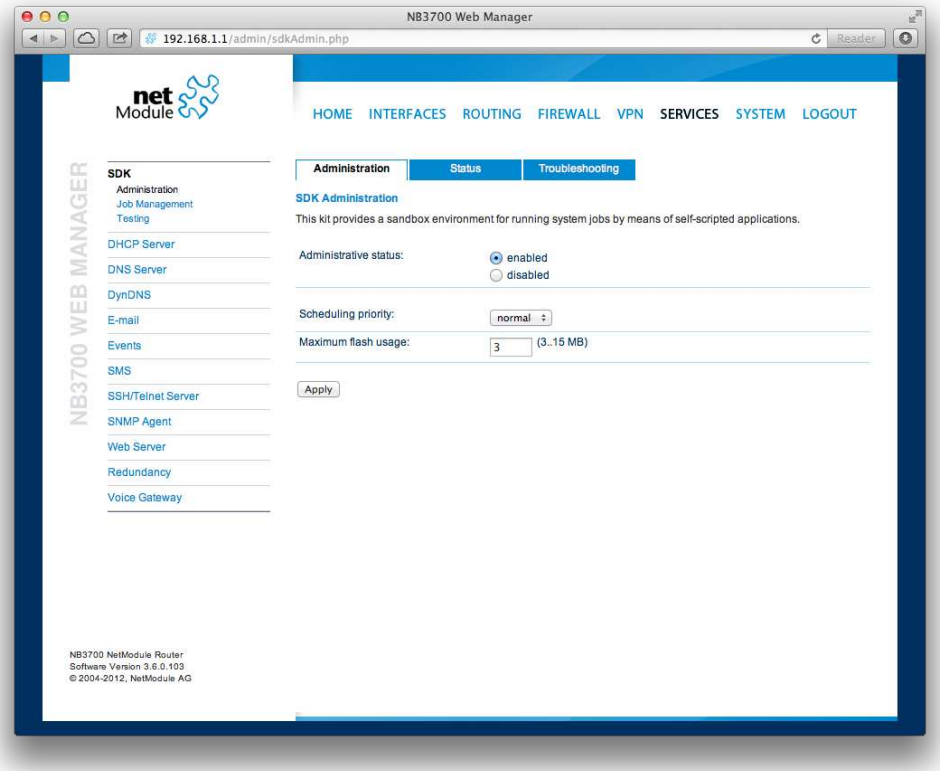


Figure 5.33.: SDK Administration

Administration

This page can be used to control the SDK host and apply the following settings:

Parameter	SDK Administration Settings
Parameter	Description
Administrative status	Specifies whether SDK scripts should run or not
Scheduling priority	Specifies the process priority of the sdkhost, higher priorities will speed up scheduling your scripts, lower ones will have less impact to the host system
Maximum flash usage	The maximum amount of MBytes your scripts can write to the internal flash
Enable watchdog	This option will enable watchdog supervision for each script which leads to a reboot of the system if the script does not respond or stopped with an exit code not equal zero.

The status page informs you about the current status of the SDK. It provides an overview about any finished jobs, you can also stop a running job there and view the script output in the troubleshooting section where you will also find links for downloading the manuals and examples.

Job Management

This page can be used to set up scripts, triggers and jobs. It is usually a good idea to create a trigger first which is made up by the following parameters:

Parameter	SDK Trigger Parameters
Name	A meaningful name to identify the trigger
Type	The type of the trigger, either time-based or event-based
Condition	Specifies the time condition for time-based triggers (e.g. hourly)
Timespec	The time specification which, together with the condition, specifies the time(s) when the trigger should be pulled
Event	The system event upon which the trigger should be pulled

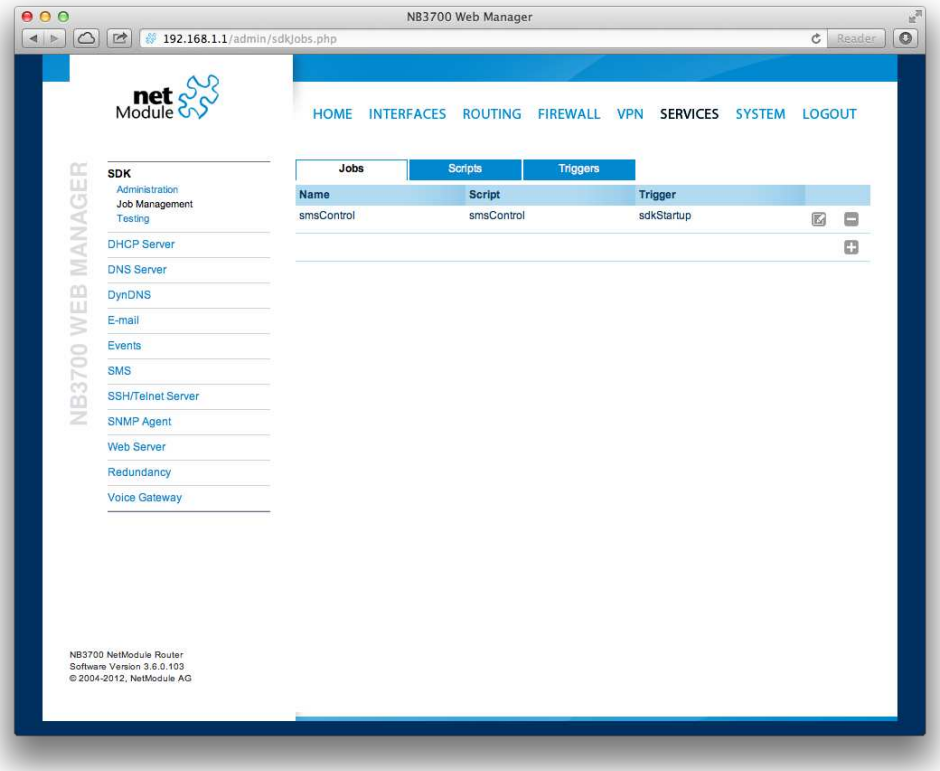


Figure 5.34.: SDK Jobs

You can now add your personal script to the system by applying the following parameters:

Parameter	SDK Script Parameters
Name	A meaningful name to identify the script
Description	An optional description of the script
Arguments	An optional set of arguments passed to the script (supports quoting)
Action	You may either edit a script, upload it to the system or select one of the example scripts or an already uploaded script

You are ready to set up a job afterwards, it can be created by using the following parameters:

Parameter	SDK Job Parameters
Name	A meaningful name to identify the job
Trigger	Specifies the trigger that should launch the job
Script	Specifies the script to be executed
Arguments	Defines arguments which can be passed to the script (supports quoting), they will precede the arguments you formerly may have assigned to the script itself

You can trigger each configured job directly which can be helpful for testing purposes.

Pages

Any programmed SDK pages will show up here.

Testing

The testing page offers an editor and an input field for optional arguments which can be used to perform test runs of your script or test dedicated portions of it. Please note that you might need to quote arguments as they will otherwise be separated by white-spaces.

```
/* arguments: 'schnick schnack "s c h n u c k"'

for (i = 0; i < argc; i++) {
    printf("argv%d: %s\n", i, argv[i]);
}

/* generates:
 *       argv0: scriptname
 *       argv1: schnick
 *       argv2: schnack
 *       argv3: s c h n u c k
 */
```

In case of syntax errors, `arena` will usually print error messages as follows (indicating the line and position where the parsing error occurred):

```
/scripts/testrun:2:10:FATAL: parse error, unexpected $, expecting ';'
```

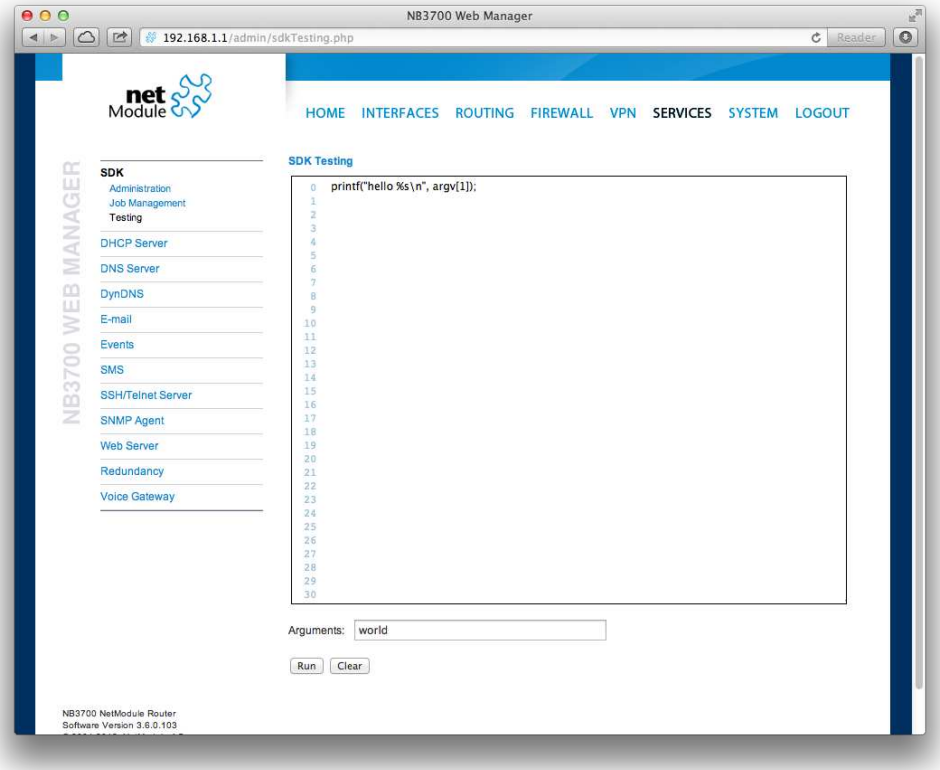


Figure 5.35.: SDK Testing

SDK Sample Application

As an introduction, you can step through a sample application, namely the SMS control script, which implements remote control over short messages and can be used to send a status of the system back to the sender. The source code is listed in the appendix.

Once enabled, you can send a message to the phone number associated with a SIM / modem. It generally requires a password to be given on the first line and a command on the second, such as:

```
admin01
status
```

We strongly recommend to use authentication in order to avoid any unintended access, however you may pass `noauth` as argument to disable it. You can then skip the first line containing the password. Having a closer look to the script, you will see that you will also be able to restrict the list of permitted senders. Please inspect the system log for troubleshooting any issues.

The following commands are supported:

Command	Action
status	Will reply a message to the sender including a short system overview
connect	Will enable the first WAN link configured on the system
disconnect	Will disable the first WAN link configured on the system
reboot	Initiates a reboot of the system
output 1 on	Turns on the first digital output port
output 1 off	Turns off the first digital output port
output 2 on	Turns on the second digital output port
output 2 off	Turns off the second digital output port

Table 5.60.: SMS Control Commands

A response to the status command typically looks like:

```
System: NB2700 hostname (00:11:22:AA:BB:CC)
WAN1: WWAN1 is up (10.0.0.1, Mobile1, UMTS, -83 dBm, LAI 12345)
GPS: lat 47.377894, lon 8.540055, alt 282.200
OVPN: client on tun0 is up (10.0.8.4)
DIO: IN1=off, IN2=off, OUT1=on, OUT2=off
```

5.7.2. DHCP Server

This section can be used to individually configure the Dynamic Host Configuration Protocol (DHCP) service for each LAN interface which will serve dynamic IP addresses to hosts in the local network. You may also have a look to the leases page where you can find an overview about negotiated client addresses.

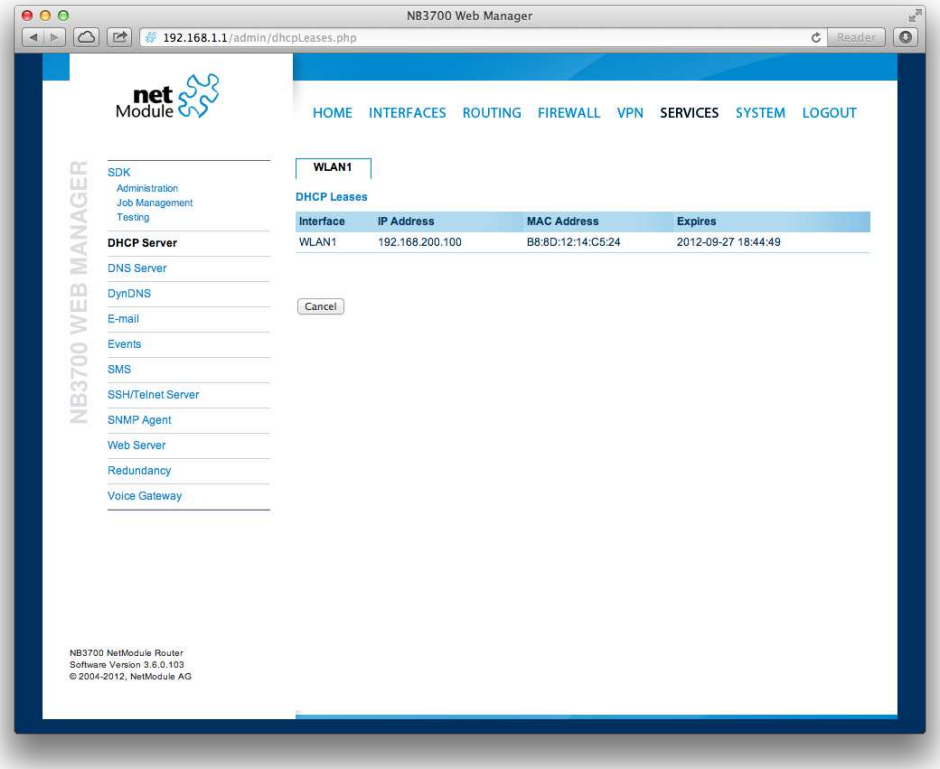


Figure 5.36.: DHCP Leases

Please note that WLAN interfaces (for each SSID) will pop up here as well in case you have configured an access point respectively.
The following settings for each interface can be applied then:

Parameter	DHCP Server Settings
Administrative status	Specifies whether the DHCP server is enabled or not
First lease address	The first address out of the range of IP addresses given to hosts
Last lease address	The last address out of this range

Parameter	DHCP Server Settings
Lease duration	Number of seconds how long a given lease shall be valid until it has to be requested again
Persistent leases	By turning on this option the router will remember issued leases even after a reboot. This can be used to ensure that the same IP address will be assigned to a particular host.
DHCP options	By default the DHCP will hand out the interface address as default gateway and the current DNS server addresses if not configured elsewise. You can specify fixed addresses here.

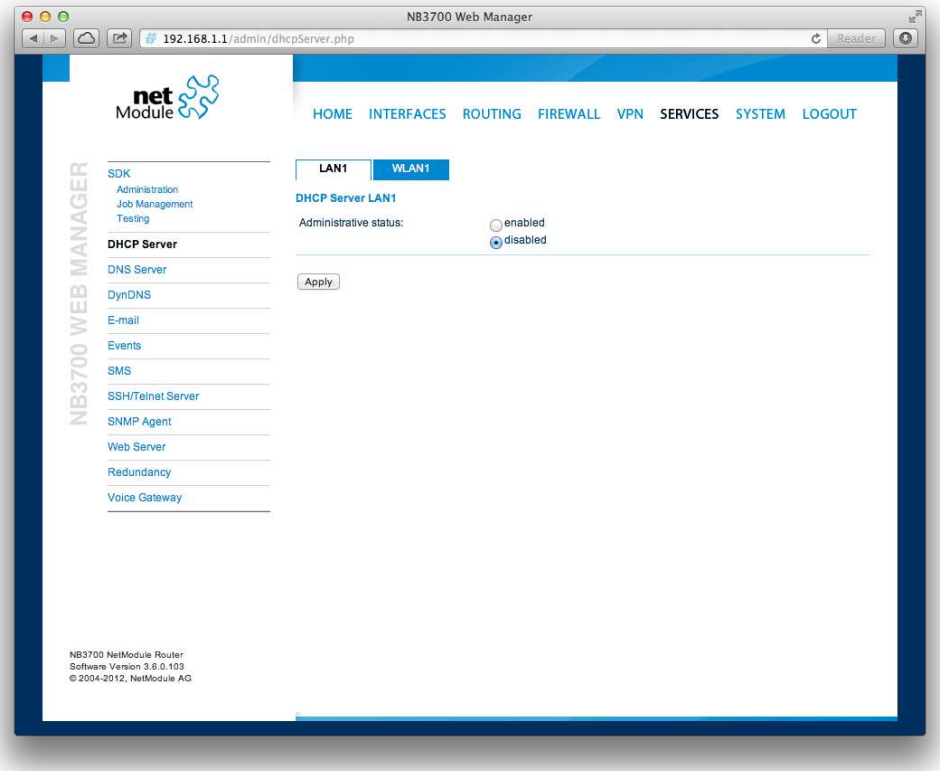


Figure 5.37.: DHCP Server

5.7.3. DNS Server

The DNS server can be used to proxy DNS requests towards servers on the net which have for instance been negotiated during WAN link negotiation. By pointing DNS requests to the router, one can reduce outbound DNS traffic as it is caching already resolved names but it can be also used for serving fixed addresses for particular host names.

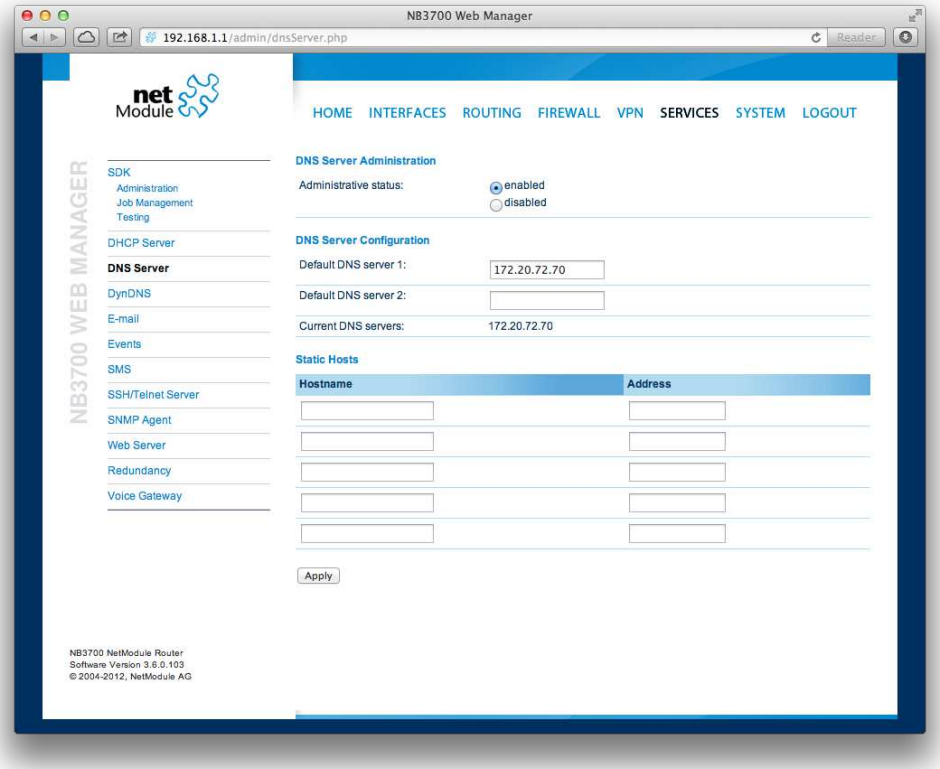


Figure 5.38.: DNS Server

The following settings can be applied:

Parameter	DNS Server Settings
Administrative status	Enables or disables the DNS server
Default DNS server 1	The primary default DNS server which will be used if no other service can be negotiated
Default DNS server 2	The secondary server which will be used in case the primary server is not available

You may further configure static hosts for serving fixed IP addresses for various hostnames. Please remember to point local hosts to the router's address for resolving them.

5.7.4. NTP Server

This section can be used to individually configure the Network Time Protocol (NTP) server function.

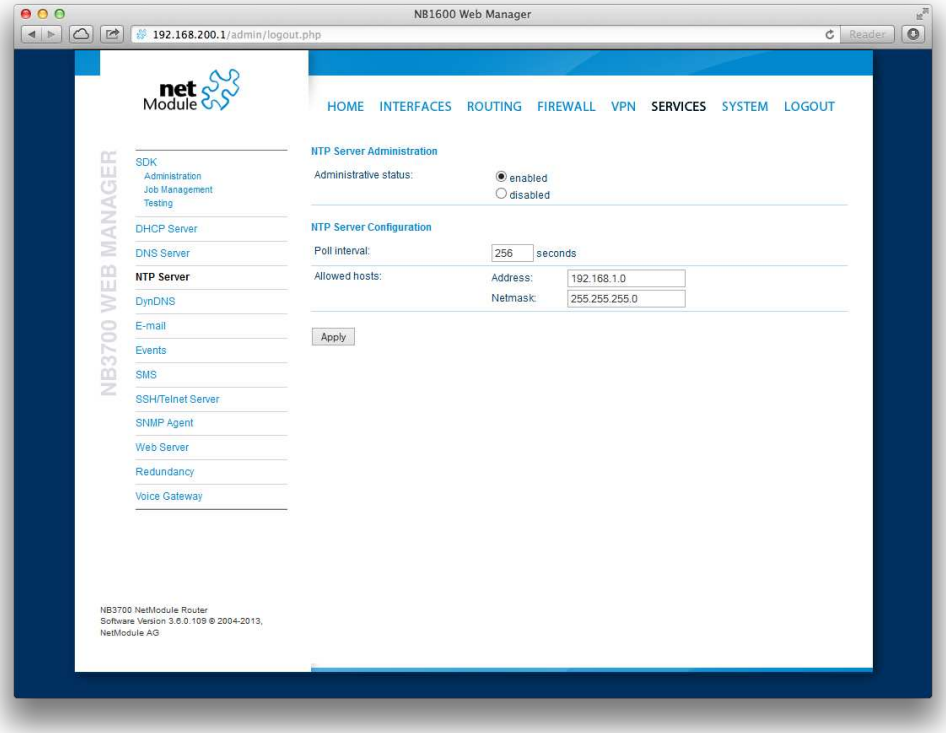


Figure 5.39.: NTP Server

The following settings for each interface can be applied then:

Parameter	NTP Server Settings
Administrative status	Specifies whether the NTP server is enabled or not
Poll interval	Defines the polling interval (64..2048 seconds) for synchronizing the time with the master clock servers
Allowed hosts	Defines the IP address range which is allowed to poll the NTP server

For setting the system time of the device see 5.8.1.

5.7.5. DynDNS

The dynamic DNS client on this box can be used to tell one or more DynDNS providers the current WAN address of this system. This address can be either derived from the current hot-link address or by querying an HTTP service in the Internet for the current Internet address. The latter might be applicable in NAT scenarios.

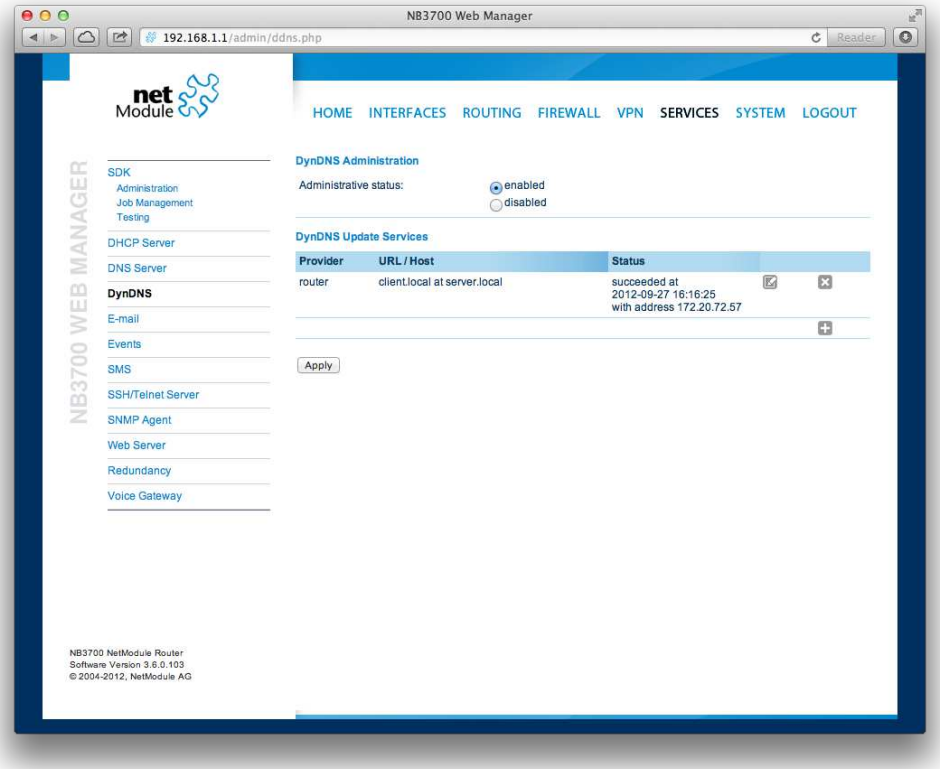


Figure 5.40.: Dynamic DNS Settings

Each service can be configured as follows:

Parameter	DynDNS Settings
Provider	You can choose one of the listed providers or provide a custom URL
Dynamic address	Specifies whether the address is derived from the hot-link or via an external service
Hostname	The host-name provided by your DynDNS service (e.g. mybox.dyndns.org)
Port	The HTTP port of the service (typically 80)

Parameter	DynDNS Settings
Username	The user-name used for authenticating at the service
Password	The password used for authentication

Please note that your NetModule router can operate as DynDNS service as well, provided that you have your hosts pointed to the DNS service of the router.

5.7.6. E-Mail

The E-Mail client can be used to send notifications to a particular E-Mail address upon certain events or by SDK scripts.

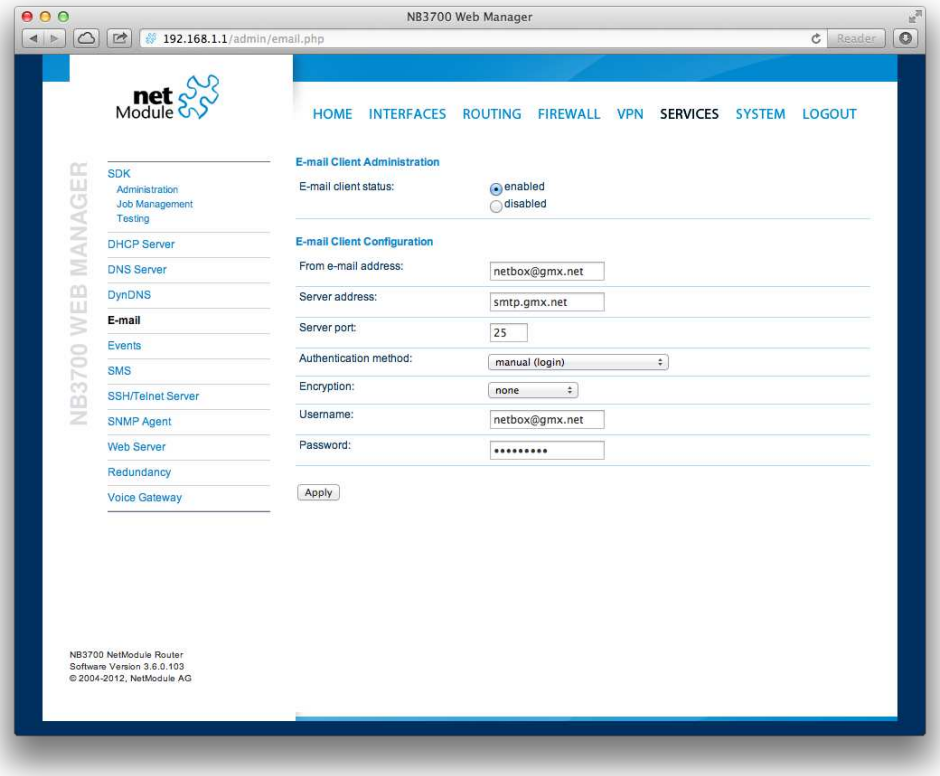


Figure 5.41.: E-Mail Settings

It can be enabled by applying the following settings.

Parameter	E-Mail Client Settings
E-mail client status	Administrative status of the E-Mail client
From e-mail address	E-Mail address of the sender
Server address	SMTP server address
Server port	SMTP server port (typically 25)
Authentication method	Select the required authentication method which will be used to authenticate against the SMTP server
Username	User name used for authentication

Parameter	E-Mail Client Settings
Password	Password used for authentication

5.7.7. Events

By using the event manager you can notify one or more recipients by SMS or E-Mail upon certain system events. The messages will contain a description provided by you and a short system info.

A list of all system events can be found in the appendix A.2.

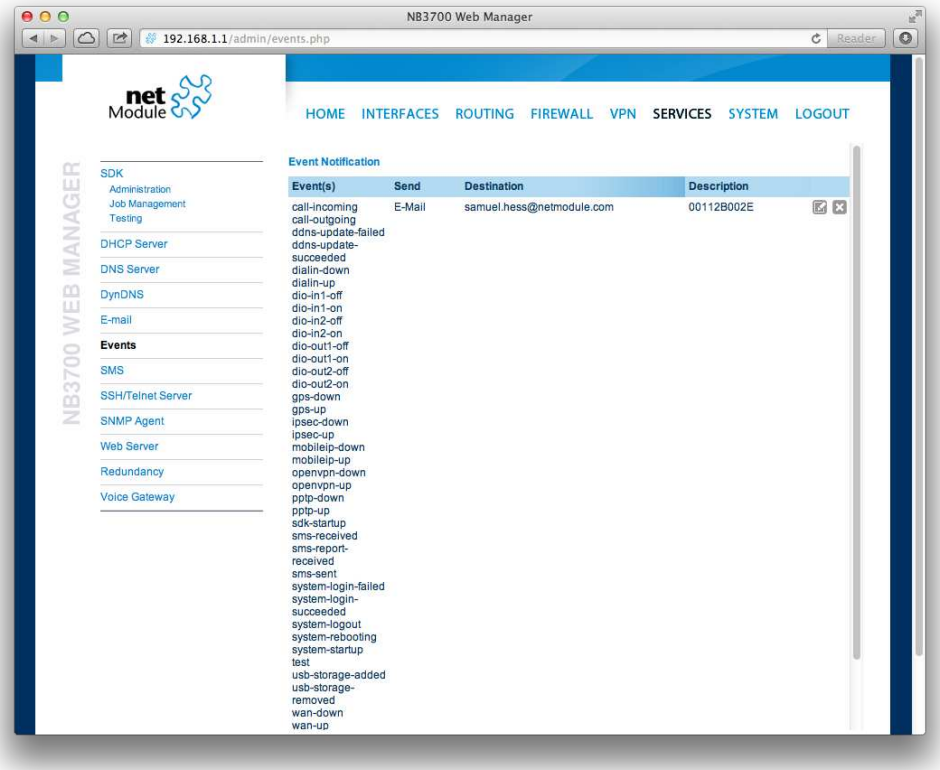


Figure 5.42.: Event Notification Settings

5.7.8. SMS

Administration

On NetModule routers it is possible to receive or send short messages (SMS) over each mounted modem (depending on the assembly options). Messages are received by querying the SIM card over a modem, so prior to that, the required assignment of a SIM card to a modem needs to be specified on the SIMs page.

Please bear in mind, in case you are running multiple WWAN interfaces sharing the same SIM, that the system may switch SIMs during operation which will also result in different settings for SMS communication.

Received messages are pulled from the SIMs and temporarily stored on the router but get cleared after a system reboot. Please consider to consult an SDK script in case you want to process or copy them.

Sending messages heavily depends on the registration state of the modem and whether the provided SMS Center service works and may fail. You may use the `sms-report-received` event to figure out whether a message has been successfully sent.

Please do not forget that modems might register roaming to foreign networks where other fees may apply. You can manually assign a fixed network (by LAI) in the SIMs section.

The relevant page can be used to enable the SMS service and specify on which it should operate.

Routing & Filtering

By using SMS routing you can specify outbound rules which will be applied whenever message are sent. On the one hand, you can forward them to an enabled modem. For a particular number, you can for instance enforce messages being sent over a dedicated SIM. Phone numbers can also be specified by regular expressions, here are some examples:

Number	Result
+12345678	Specifies a fixed number
+1*	Specifies any numbers starting with `+1`
+1*9	Specifies any numbers starting with `+1` and ending with `9`
+[12]*	Specifies any numbers starting with either `+1` or `2`

Table 5.66.: SMS Number Expressions

Please note that numbers have to be entered in international format including a valid

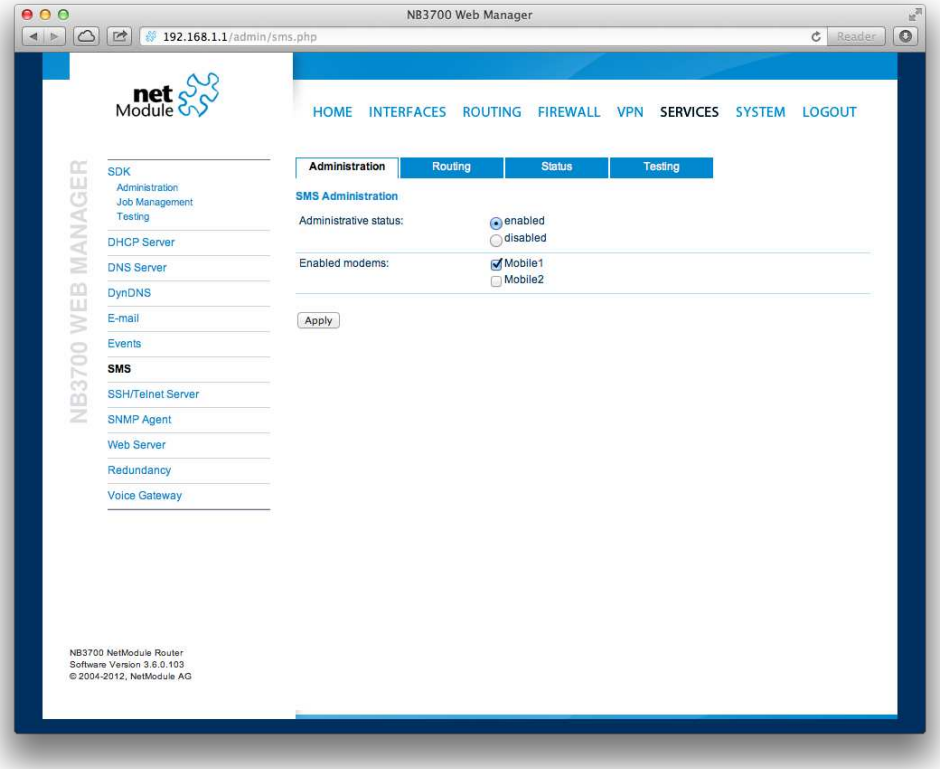


Figure 5.43.: SMS Configuration

prefix.
On the other hand, you can also define rules to drop outgoing messages, for instance, when you want to avoid using any expensive service or international numbers.

Both types of rules form a list will be processed by order, forwarding outgoing messages over the specified modem or dropping them. Messages which are not matching any of the rules below will be dispatched to the first available modem.

Filtering serves a concept of firewalling incoming messages, thus either dropping or allowing them on a per-modem basis. The created rules are processed by order and in case of matches will either drop or forward the incoming message before entering the system. All non-matching messages will be allowed.

Status

The status page can be used to the current modem status and get information about any sent or received messages. There is a small SMS inbox reader which can be used to view or delete the messages. Please note that the inbox will be cleared each midnight in case it exceeds 512 kBytes of flash usage.

Testing

This page can be used to test whether SMS sending in general or filtering/routing rules works. The maximum length per message part is limited to 160 characters, we also suggest to exclusively use characters which are supported by the GSM 7-bit alphabet.

5.7.9. SSH/Telnet Server

Apart from the Web Manager, the SSH and Telnet services can be used to log into the system. Valid users include *root* and *admin* as well as additional users as they can be created in the User Accounts section. Please note, that a regular system shell will only be provided for the *root* user, the CLI will be launched for any other user whereas normal users will only be able to view status values, the *admin* user will obtain privileges to modify the system.

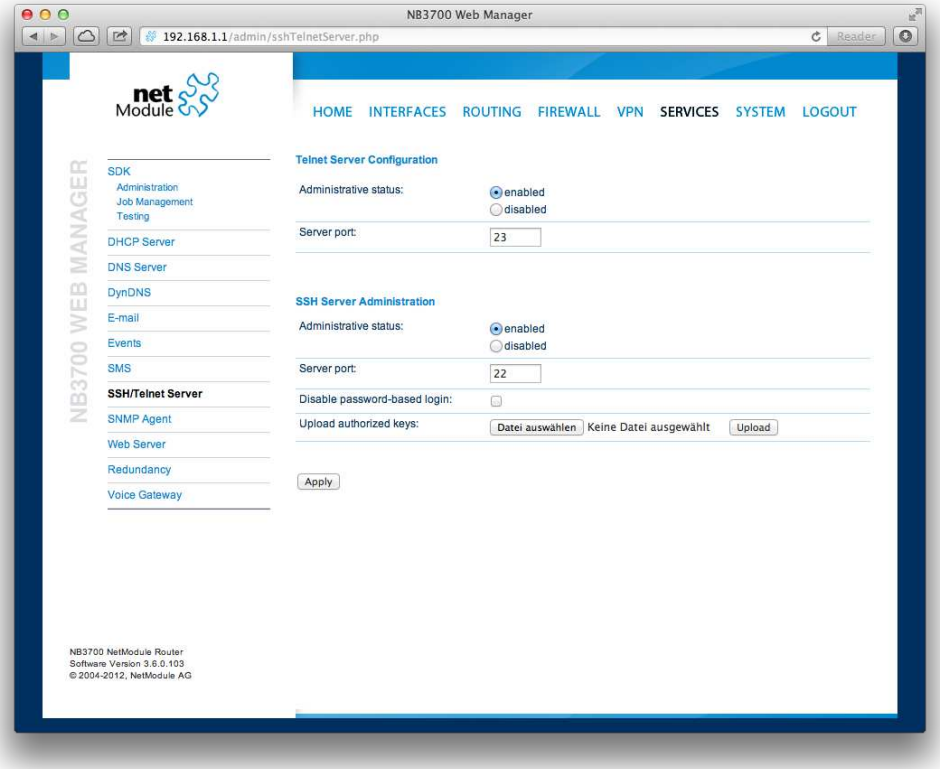


Figure 5.44.: SSH and Telnet Server

Please note that these services will be accessible from the WAN interface also. In doubt, please consider to disable or restrict access to them by applying applicable firewall rules. The following parameters can be applied to the Telnet service:

Parameter	Telnet Server Settings
Administrative status	Whether the Telnet service is enabled or disabled
Server port	The TCP port of the service (usually 23)

The following parameters can be applied to the SSH service:

Parameter	SSH Server Settings
Administrative status	Whether the SSH service is enabled or disabled
Server port	The TCP port of the service (usually 22)
Disable password-based login	By turning on this option, all users will have to authenticate by SSH keys which can be uploaded to the router.

5.7.10. SNMP Agent

NetModule routers are equipped with an SNMP daemon, supporting basic MIB tables (such as ifTable), plus additional enterprise MIBs to manage multiple systems. Our VENDOR-MIB is listed in the appendix or can be downloaded directly from the router.

The VENDOR-MIB tables offer some additional information over the system and its WWAN, GNSS and WLAN interfaces. They can be accessed over the following OIDs:

Parameter	Vendor MIB OID Assignment
NBAdminTable	.1.3.6.1.4.1.31496.10.40
NBWwanTable	.1.3.6.1.4.1.31496.10.50
NBGnssTable	.1.3.6.1.4.1.31496.10.51
NBDioTable	.1.3.6.1.4.1.31496.10.53
NBWlanTable	.1.3.6.1.4.1.31496.10.60

They offer facilities for:

- rebooting the device
- updating to a new system software via FTP/TFTP/HTTP
- updating to a new system configuration via FTP/TFTP/HTTP
- getting WWAN/GNSS/WLAN/DIO information

Typical SNMP Commands

Setting MIB values and triggering extensions is generally limited to the SNMPv3 admin user. It is possible to specify an administrative host for SNMP v1/2c.

The SNMP extensions can be read and triggered as follows:

Listing 5.1: Getting the software version of the system:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.1.0
```

Listing 5.2: Getting the kernel version:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.2.0
```

Listing 5.3: Getting the serial number:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.3.0
```

Listing 5.4: Restarting the device:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.10.0 i 1
```

Listing 5.5: Running a configuration update:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.11.0 s
"http://server/directory"
```

You can use TFTP, HTTP, HTTPS and FTP URLs, specifying a username/password or a port is not yet supported. Please note that config updates expect a zip-file named `<serial-number>.zip` in the specified directory.

Listing 5.6: Getting the configuration update status:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.12.0
```

The return value can be one of: `succeeded (1)`, `failed (2)`, `inprogress (3)`, `notstarted (4)`.

Listing 5.7: Running a software update:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.13.0 s
"http://server/directory"
```

Listing 5.8: Getting the software update status:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.3.6.1.4.1.31496.10.40.14.0
```

The return value can be one of: `succeeded (1)`, `failed (2)`, `inprogress (3)`, `notstarted (4)`.

Listing 5.9: Setting digital OUT1:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  .1.3.6.1.4.1.31496.10.53.10.0 i 0
```

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  .1.3.6.1.4.1.31496.10.53.10.0 i 1
```

Listing 5.10: Setting digital OUT2:

```
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  .1.3.6.1.4.1.31496.10.53.11.0 i 0
snmpset -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  .1.3.6.1.4.1.31496.10.53.11.0 i 1
```

Listing 5.11: Listing discovered devices:

```
snmpget -v 3 -u admin -n "" -l authNoPriv -a MD5 -x DES -A ↩
   admin01admin01 192.168.1.1  1.0.8802.1.1
```

5.7.11. SNMP Configuration

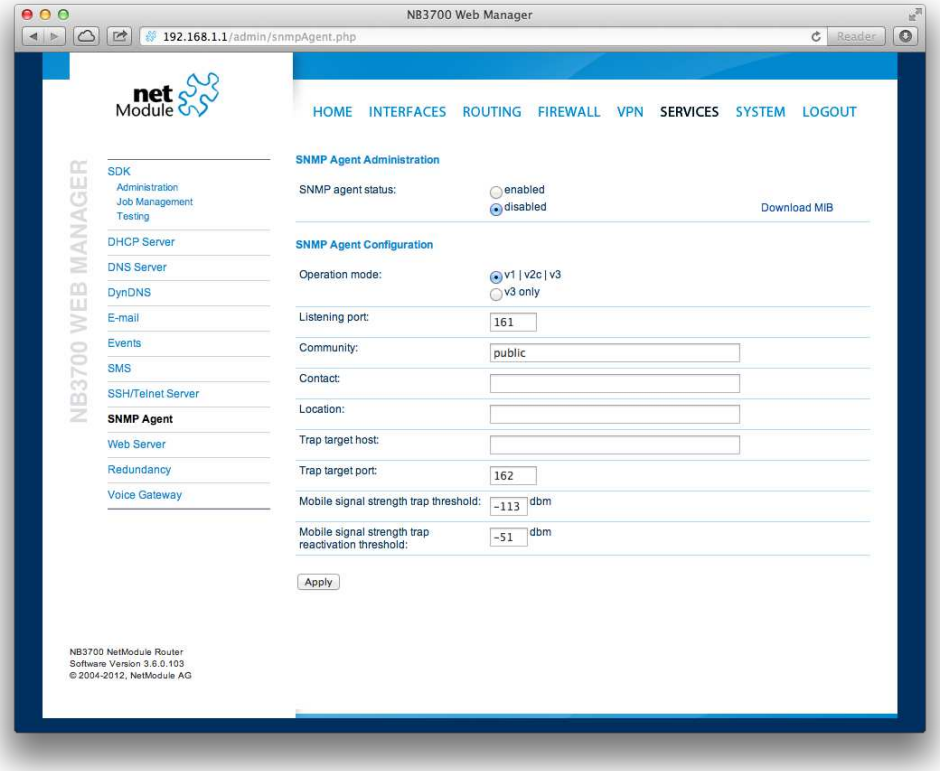


Figure 5.45.: SNMP Agent

The following parameters can be used to configure the SNMP agent:

Parameter	SNMP Configuration
Administrative status	Enable or disable the SNMP agent
Operation mode	Specifies if agent should run in compatibilty mode or for SNMPv3 only
Contact	System maintainer or other contact information
Location	Location of the device
Listening Port	SNMP agent port

Once the SNMP agent is enabled, SNMP traps can be generated using SDK scripts.

5.7.12. SNMP Authentication

When running in SNMPv3, it is possible to configure the following authentication settings:

Parameter	SNMPv3 Authentication
Authentication	Defines the authentication (MD5 or SHA)
Encryption	Defines the privacy protocols to use (DES or AES

In general, the admin user can read and write any values. Read access will be granted to any other system users.
There is no authentication/encryption in SNMPv1/v2c and should not be used to set any values. However, it is possible to define its communities and authoritive host which will be granted administrative access.

Parameter	SNMPv1/v2c Authentication
Read community	Defines the community name for read access
Admin community	Defines the community name for admin access
Allowed host	Defines the host which is allowed for admin access

Attention must be paid to the fact that SNMP passwords have to be more than 8 characters long. Shorter passwords will be doubled for SNMP (e.g. `admin01` becomes `admin01admin01`).
Please note that the SNMP daemon is also listening on WAN interfaces and it is therefore suggested to restrict the access with the firewall.

5.7.13. Web Server

This page can be used to configure different ports for accessing the Web Manager via HTTP/HTTPS. We strongly recommend to use HTTPS when accessing the web service via a WAN interface as the communication will be encrypted and thus avoids any misuse of the system.

In order to enable HTTPS you would need to generate or upload a server certificate in the section 5.8.6.

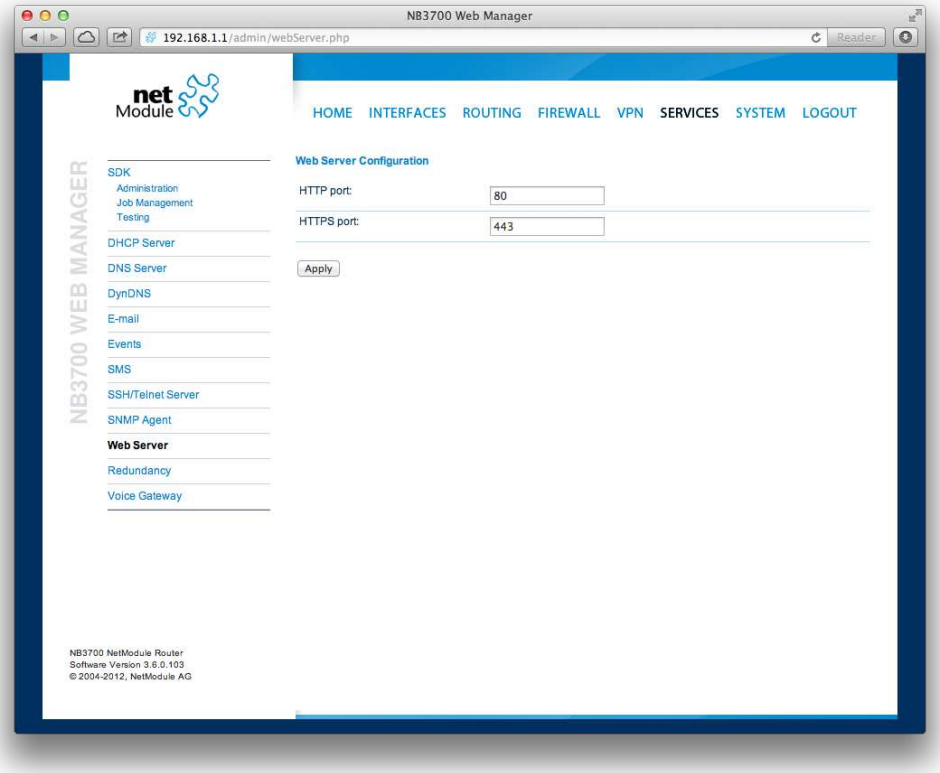


Figure 5.46.: Web Server

Parameter	Web Server Settings
Administrative Status	Enableor disable the Web server
HTTP port	Web server port for HTTP connections
HTTPS port	Web server port for HTTPS connections
Enable CLI-PHP	Enable CLI-PHP service (see chapter 6.15)

5.7.14. Redundancy

This page can be used to set up a redundant pair of NetModule routers (or other systems) by running the Virtual Router Redundancy Protocol (VRRP) between them. A typical VRRP scenario defines a first host playing the master and another the backup device, they both define a virtual gateway IP address which will be distributed by gratuitous ARP messages for updating the ARP cache of all LAN hosts and thus redirecting the packets accordingly. A takeover will happen within approximately 3 seconds as soon as the partner is not reachable anymore (checked via multicast packets). This may happen when one device is rebooting or the Ethernet link went down. Same applies when the WAN link goes down.

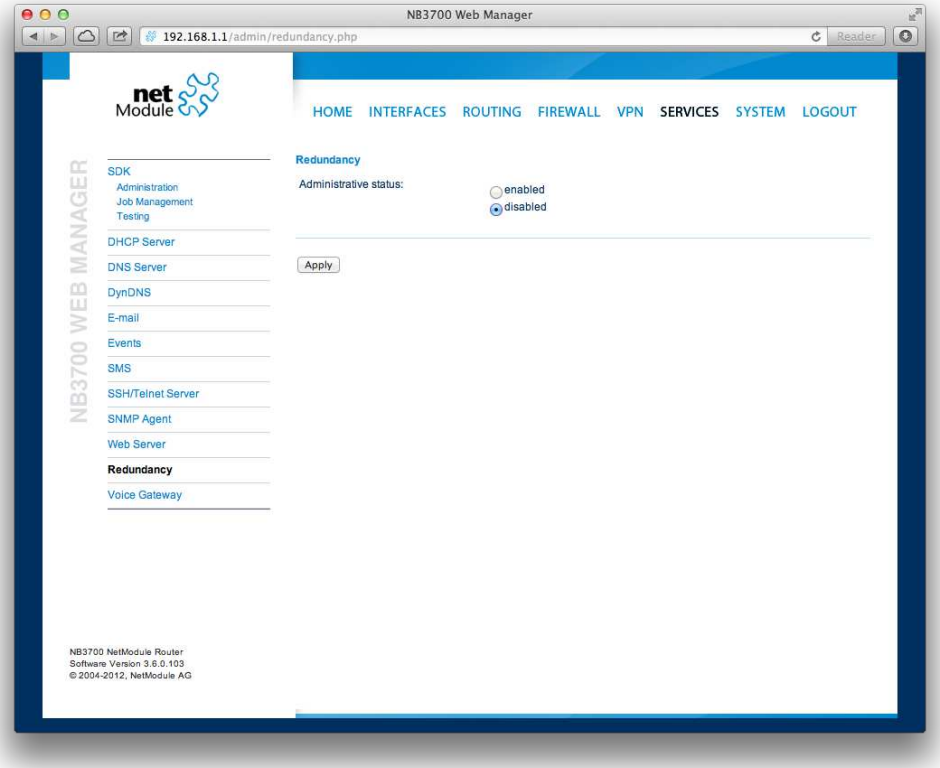


Figure 5.47.: VRRP Configuration

In case DHCP has been activated, please keep in mind that you will need to reconfigure the DHCP gateway address offered by the server and let them point to the virtual gateway address. In order to avoid conflicts you may turn off DHCP on the backup device or even better, split the DHCP lease range across both routers in order to prevent any lease duplication.

Parameter	Redundancy Configuration
Administrative status	Administrative status
Role	The role of this system (either master or backup)
VID	The Virtual Router ID (you can theoretically run multiple instances)
Interface	Interface on which VRRP should be performed
Virtual gateway address	The virtual gateway address formed by the participating hosts

We assign a priority of `100` to the master and `1` to the backup router. Please adapt the priority of your third-party device appropriately.

5.7.15. Voice Gateway

Depending on your hardware, you can set up a voice gateway on the router which can be connected by any VoIP client from the local network capable of the SIP protocol. It hereby listens for arriving SIP calls and forwards them as a GSM call on the modem which has been configured. Due to this nature only one concurrent call is possible.

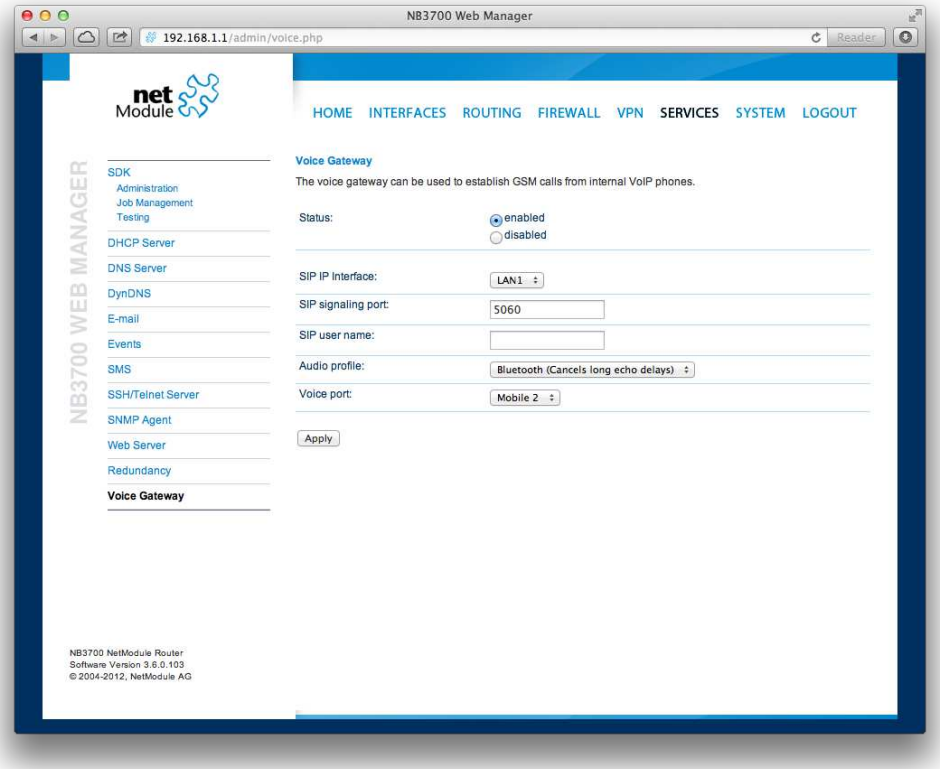


Figure 5.48.: Voice Gateway

The following parameters can be used to set it up:

Parameter	Voice Gateway Settings
Administrative status	Specifies whether the gateway shall be enabled or disabled
SIP interface	Specifies the local interface (LAN or WLAN) to which should be listened for incoming calls
SIP port	Specifies the port on which should be listened
SIP user name	reserved for future use

Parameter	Voice Gateway Settings
Audio profile	Selects the audio profile which should be applied to outgoing calls. This parmeter influences echo cancelation. For nomal use select *Bluetooth*
Voice port	Selects the modem on which GSM calls shall be established

Please bear in mind, in case you are running multiple WWAN interfaces sharing the same SIM, that the system may switch SIMs during operation which will also result in different settings for voice communication.

Client Configuration

The sip client should be configured to use the router as a voice gateway. The easiest way to achieve this is to configure the router as proxy. The Voice Gateway does not require authenticationi however it may be necessary to fill in dummy values as user ID, Domain and Password. Any SIP client with access to the *SIP IP Interface* can use the router as a voice gateway.

Sample configuration for the Counter Path X-Lite client (Version 5.0.0 build 67284)

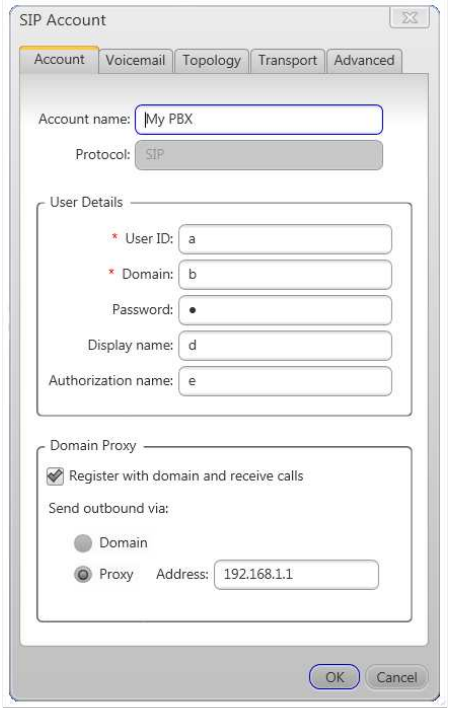


Figure 5.49.: Voice Client Configuration

5.8. SYSTEM

5.8.1. System

System Settings

The following system parameters can be set:

Parameter	System Settings
Local hostname	The hostname of the system
Application area	The desired application area which influences the system behaviour such as registration timeouts or other adaptions when operating in mobile enviroments.
Syslog redirect address	Specifies an IP address to which system log messages should be redirected to. A tiny system log server for Windows is included in TFTP32 which can be downloaded from our website.

Parameter	System Settings
Syslog max. file size	The maximum size of message log files in kilobytes until they will be rotated
Reboot delay	The number of seconds which will be waited before regular system reboots (might be needed for `system-rebooting` events)
Enable discovery	Enables host discovery over LLDP or CDP. Discovered neighbors can be found on the LAN status page or via SNMP.
Banks to be displayed	You can configure the behavior of the status LEDs on the front panel of your device. They are usually divided into two banks (top/bottom) and are either indicating the connection status or the digital IO port status. You may configure toggle mode, so that the LEDs periodically cycle between the two states.

Time & Region

This page can be used for setting the system time and configuring the time zone. You may further enable daylight saving changes (e.g. automatically switching from summer to winter time) for your specific time zone.

NetModule routers can synchronize their system time by using one or more servers by the help of the Network Time Protocol (NTP) or via GPS. If enabled, the time synchronization is usually triggered after a WAN link has come up but before starting any VPN connections. Further time synchronization cycles are scheduled in background.

Parameter	Time & Region
Time Synchronisation	Enable/disable time synchronization
NTP server	Address of the primary NTP server
NTP server 2	Optionally, the address of a second NTP server
Sync time from GPS	Derive time from first GPS device (if enabled)

Reboot

This page can be used to set up a periodic automatic reboot but also to trigger a manual reboot which will be issued immediately.

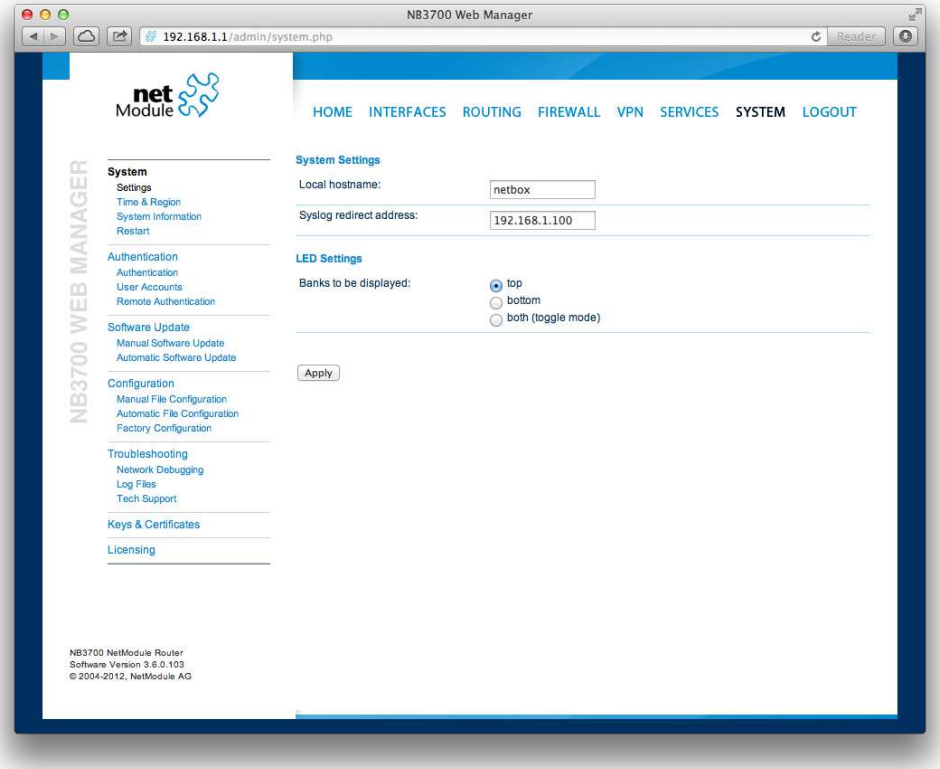


Figure 5.50.: System

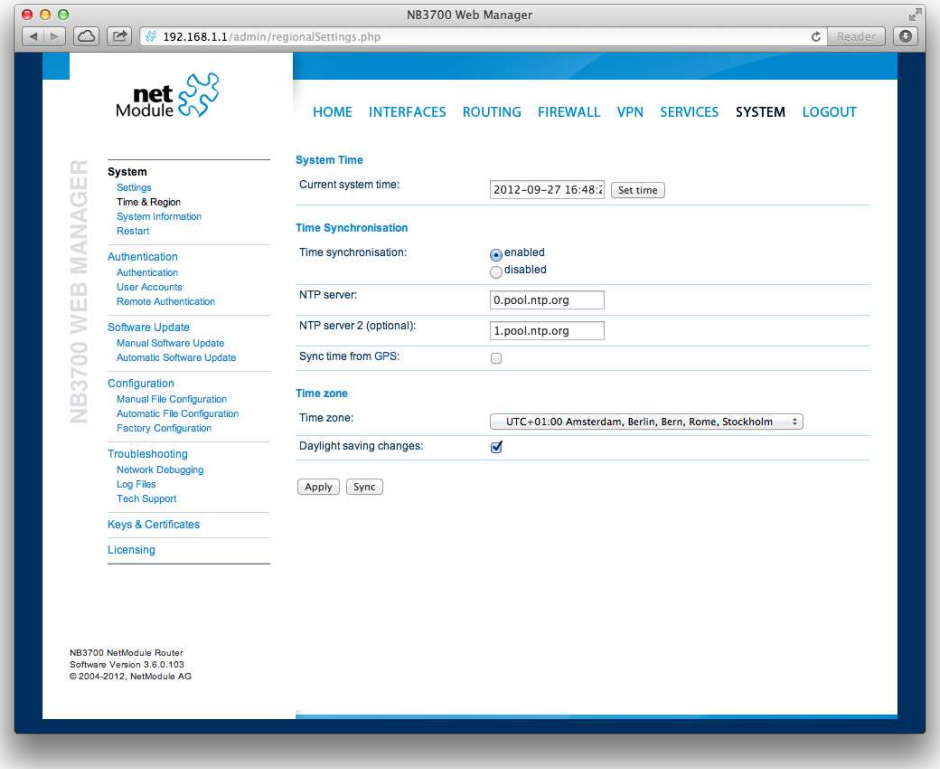


Figure 5.51.: Regional settings

5.8.2. Authentication

This pages offers a simple shortcut to only allow secure connections (SSH, HTTPS) for managing the router.

User Accounts

By using this page you can manage the user accounts on the system. The standard `admin` user is a built-in power user that has permission to access the Web Manager and other administrative services and is used by several services as default user. Keep in mind that the `admin` password will be also applied to the `root` user which is able to enter a system shell.

Any other user represents a user with lower privileges, for instance it has only permission to view the status page or retrieve status values when using the CLI.

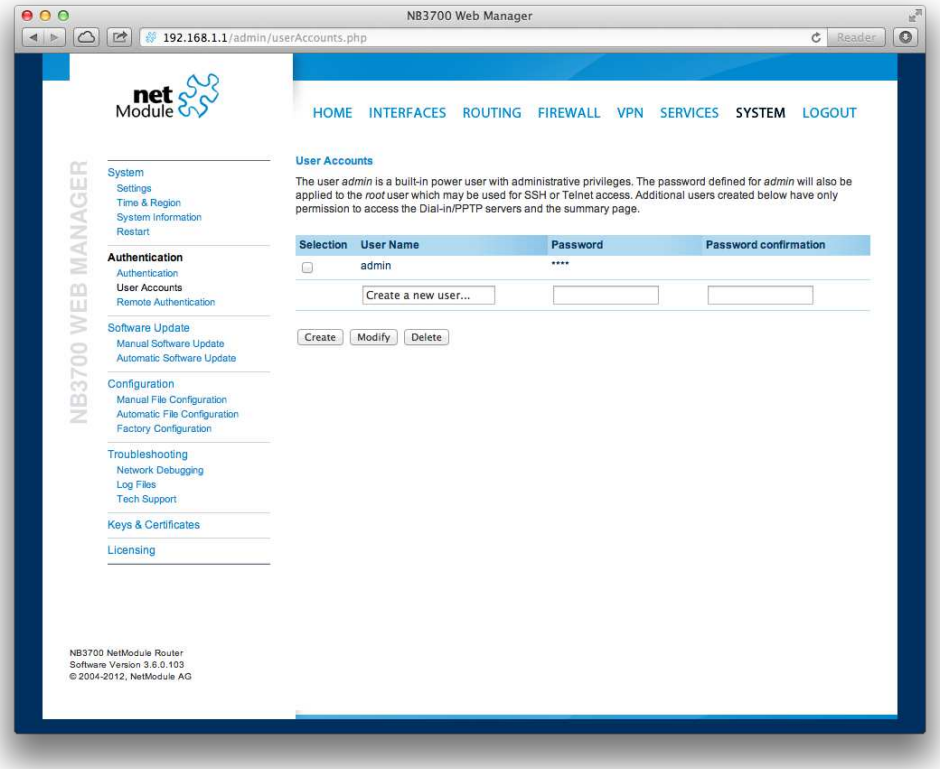


Figure 5.52.: User Accounts

Parameter	User accounts management
User name	The name of the user (avoid whitespaces or special chars)

Parameter	User accounts management
Password	The password of the user
Password confirmation	The confirmed password of the user

You will be able to modify or delete existing users here as well.

Remote Authentication

A RADIUS server can be used for authenticating remote users. This applies for the Web Manager, the WLAN network and other services supporting and incorporating remote authentication.

It can be configured as follows:

Parameter	Remote authentication settings
Administrative status	Defines whether a remote server should be used for authentication
RADIUS server	The RADIUS server address
RADIUS secret	The secret used to authenticate against the RADIUS server
Authentication port	The port used for authentication
Accounting port	The port used for accounting messages
Use for login	This option enables remotely-defined users to access the Web Manager, otherwise it is only used by services which have explicitly configured it (e.g. WLAN)

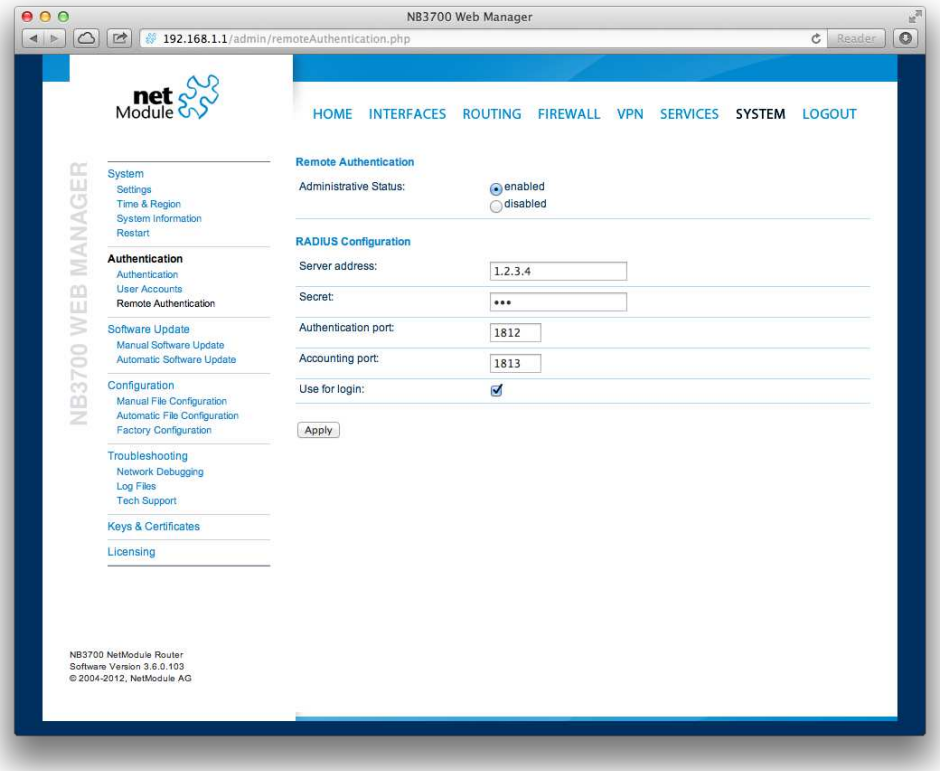


Figure 5.53.: Remote Authentication

5.8.3. Software Update

Manual Software Update

This menu can be used to run a manual software update of the system.

Parameter	Manual Software Update
Update operation	The update operation method being used. You can upload the image, download it from an URL or use the latest version from our server
URL	The server URL where the software update image should be downloaded from. Supported protocols are TFTP, HTTP, HTTPS, and FTP. Provide a URL like `protocol://server/path/file`

When issuing a software update, the current configuration (including files like keys/certificates) will be backuped. Any other modifications to the filesystem will be erased.

The configuration is generally backward-compatible. We also apply forward compatibility when downgrading to a previous software within the same release line, which is accomplished by sorting out unknown configuration directives which actually may lead to loss of settings and features. Therefore, it's always a good idea to keep a copy of the working configuration.

Attention: In case you perform a major downgrade with a previous release line (e.g. 3.7.0 to 3.6.0), please ensure to always use the latest release of that branch (i.e. 3.6.0.X) as only those tend to be fully forward-compatible. Also keep in mind, that some hardware features may not work (e.g. if not implemented in that version). In doubt, please consult our support team.

A software image can be either uploaded via the Web Manager or retrieved from a specific URL. It will be unpacked and deployed to a spare partition which gets activated if the update completed successfully. The whole procedure is accompanied by all green LEDs flashing up, the subsequent system reboot gets denoted by a slowly blinking Status LED. The backuped configuration will be applied at bootup and the Status LED will blink faster during this operation. Depending on your configuration, this may take a while.

Automatic Software Update

This menu can be used to run a automatic software update of the system.

Parameter	Automatic software update
Status	Enable/disable automatic software update

Parameter	Automatic software update
Time of day	Every day at this time the router will do a check for updates
URL	The server URL where the software update package should be downloaded from. Supported protocols are TFTP, HTTP, HTTPS, and FTP. Provide a URL like `protocol://server/path/file`

Remark: SSL certificates of HTTPS URLs will be only verified if a list of CA root certificates are provided under 5.8.6.

After the new software has been installed, the latest running configuration will be applied afterwards during bootup. This is indicated by a faster green blinking of the Status LED.

5.8.4. Configuration

Configuration via the Web Manager becomes tedious for larger volumes of devices. The router therefore offers automatic and manual file-based configuration to automate things. Once you have successfully set up the system you can back up the configuration and restore the system with it afterwards. You can either upload a single configuration file (.cfg) or a complete package (.zip) containing the configuration file and a packed version of other essential files (such as certificates) in the root directory.

Manual File Configuration

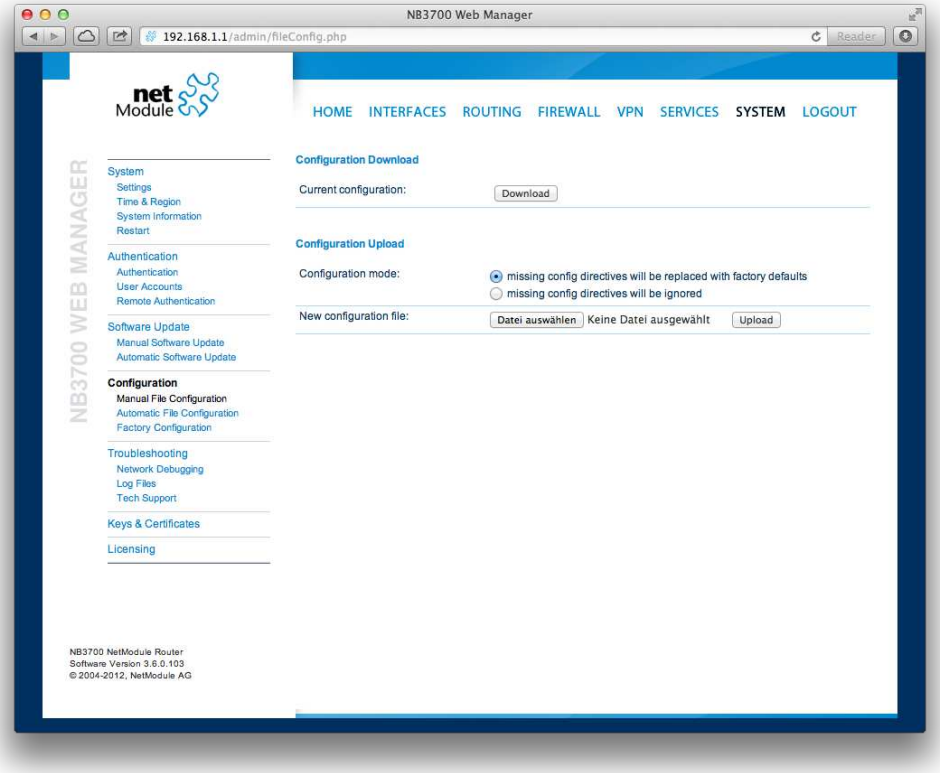


Figure 5.54.: Manual File Configuration

This section can be used to download the currently running system configuration (including essential files such as certificates). In order to restore a particular configuration you can upload a configuration previously downloaded. You can choose between missing configuration directives set to factory defaults or getting ignored, that means, potentially existing configuration directives will be kept at the system.

Automatic File Configuration

This menu can be used to run an automatic configuration update of the system. It is configured as follows:

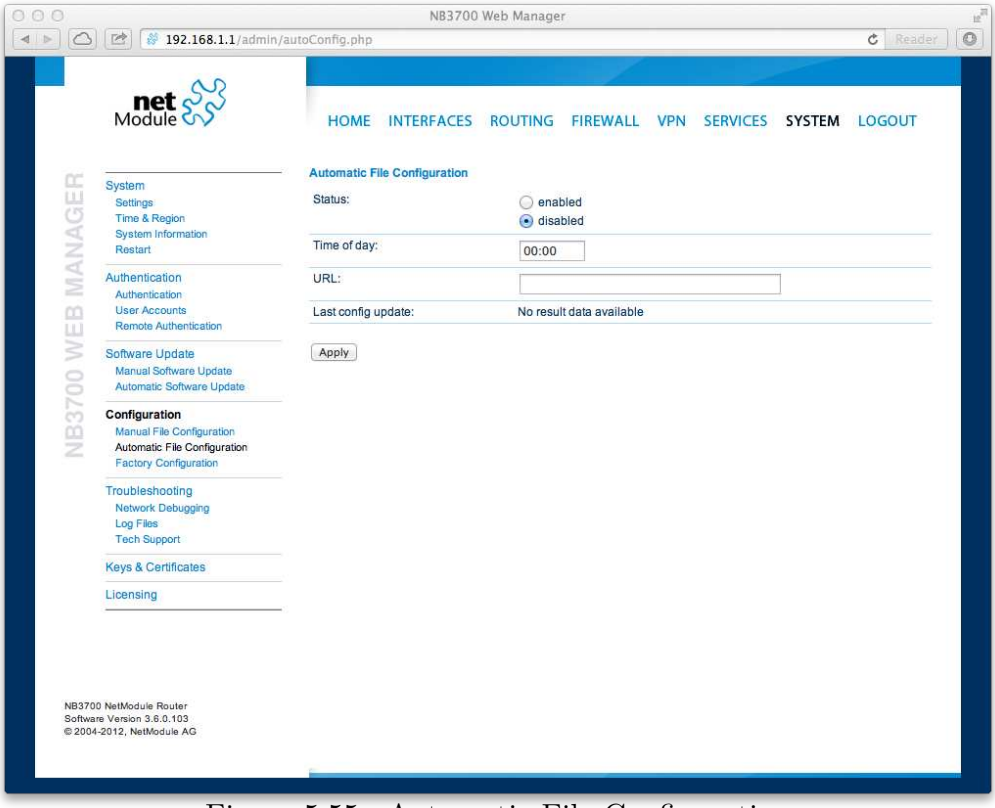


Figure 5.55.: Automatic File Configuration

Parameter	Automatic File Configuration
Status	Enable/disable an automatic configuration update
Time of day	Time of day when the system should check for updates
URL	The URL where the configuration file should be retrieved from (supported protocols are HTTP, HTTPS, TFTP, FTP)

Factory Configuration

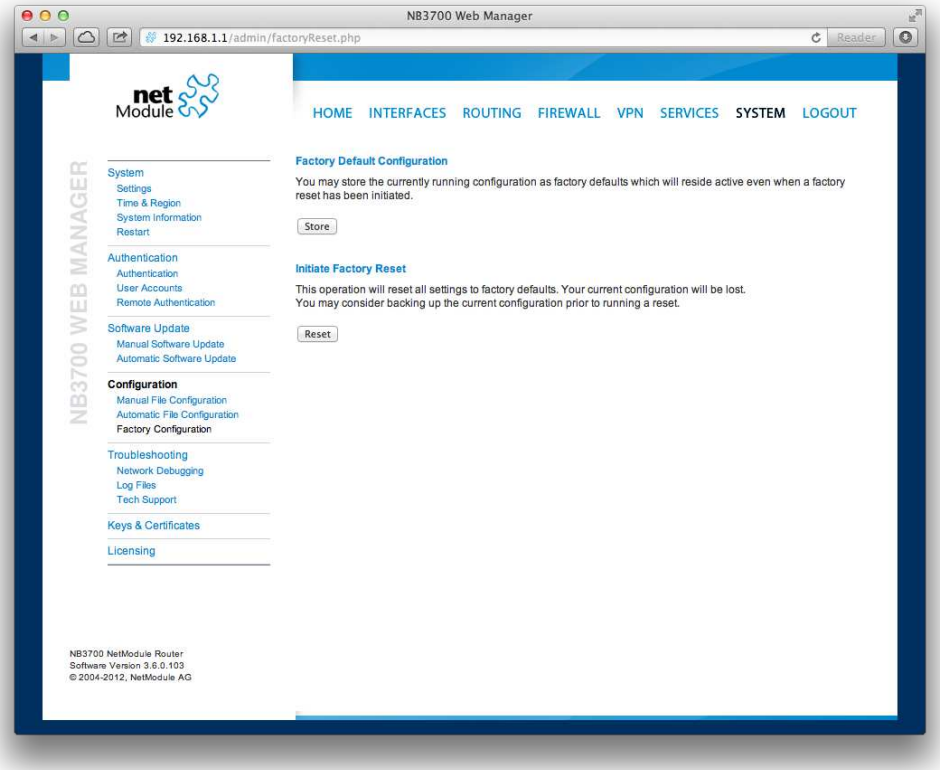


Figure 5.56.: Factory Configuration

This menu can be used to reset the device to factory defaults. Your current configuration will be lost. This procedure can also be initiated by pressing and holding the *Reset* button for at least five seconds. A successfully initiated factory reset can be noticed by all LEDs having been turned on. The factory reset will set the IP address of the first Ethernet interface back to 192.168.1.1. You will be able to communicate again with the device using the default network parameters. You may store the currently running

configuration as factory defaults which will reside active even when a factory reset has been initiated (e.g. by your service staff).

Please ensure that this corresponds to a working configuration. A real factory reset to the default settings can be achieved by restoring the original factory configuration and initiating the factory reset again.

5.8.5. Troubleshooting

Network Debugging

Log Files

You can view the system log here by selection the option *Debug log* or if you are interested in the boot log select *Boot log*.

Another way to see what is going on on the box is opening a SSH or Telnet session as *root* and typing `tail-log`. Furthermore the system log can be redirected to a syslog server, see section 5.8.1.

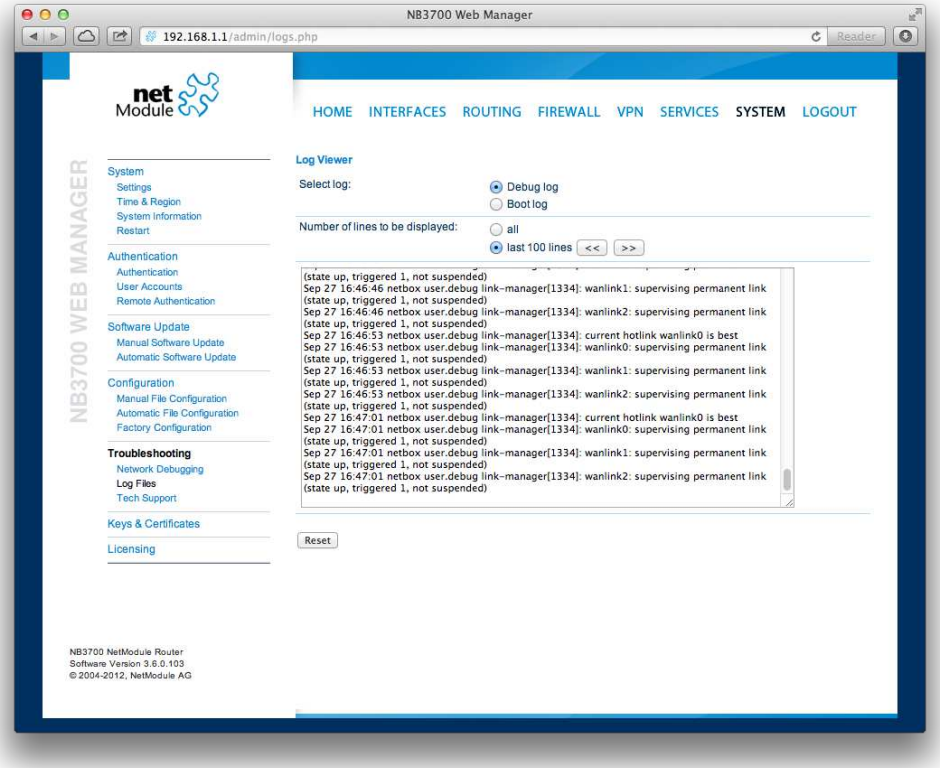


Figure 5.57.: Log Viewer

Tech Support

You can generate and download a tech support file here. We strongly recommend providing this file when getting in touch with our support team, either by e-mail or via our on-line support form, as it would significantly speed up the process of analyzing and resolving your problem. Log files can be viewed a downloaded and reset here. Please study them carefully in case of any issues. Various tools reside on this page for further analysis of potential configuration issues.

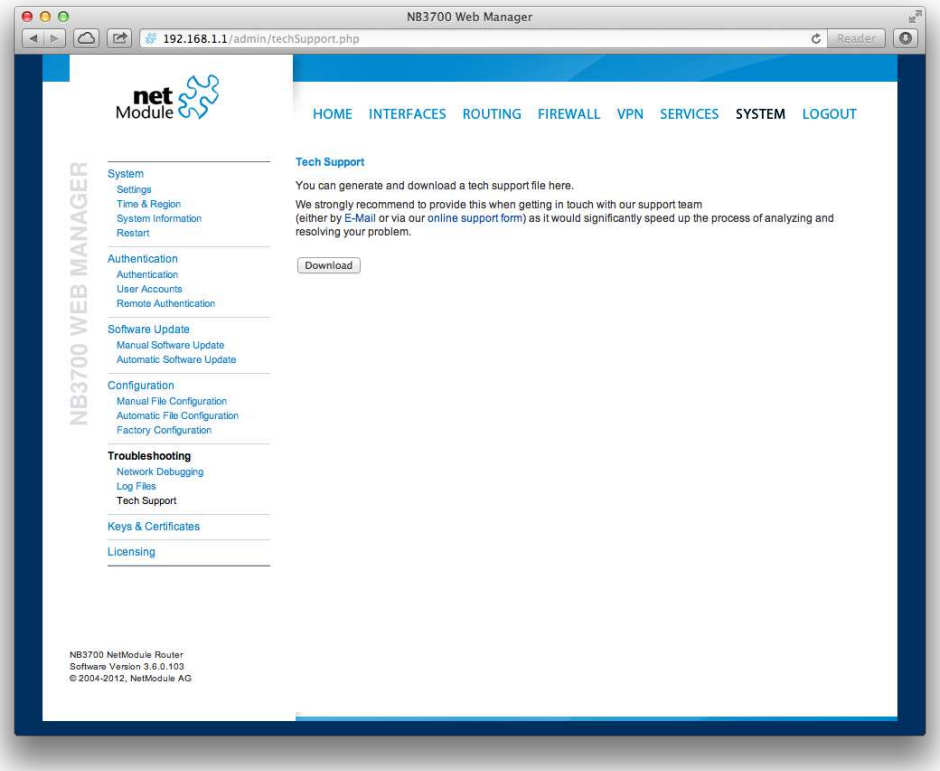


Figure 5.58.: Tech Support File

It is possible to trace any IP interface and inspect individual packet flows between hosts. This can be achieved by logging onto the box and start a network packet capture by using the tool *tcdump*. We recommend to use the -n switch to bypass name resolution (e.g. `tcpdump -n -i lan0`). You may also generate a dump in PCAP format using the Web Manager, download it to your computer and perform further inspections with Wireshark (available at www.wireshark.org).

5.8.6. Keys and Certificates

The key and certificate page lets you generate required files for securing your services (such as the HTTP and SSH server).

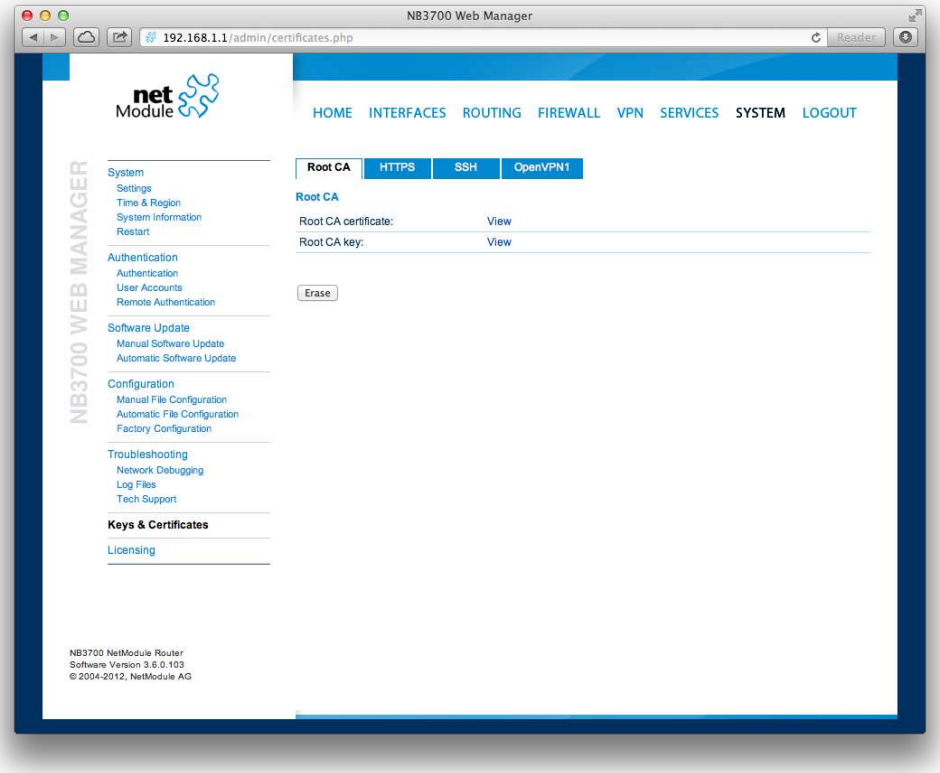


Figure 5.59.: Keys and certificates management

The following terms are used:

Term	Description
Root CA	The root Certificate Authority (CA) which issues certificates, its key can be used to certify it at trusted third party on other systems
Certificate	Corresponds to a digital certificate which uses a signature to bind a public key with an identity
Key	Corresponds to an either public or private key
CSR	Certificate Signing Request, which can be used to sign a certificate by a third party authority

Term	Description
P12	PKCS12 container format which can include certificates and keys protected by passphrase
RSA	An encryption algorithm based on the fact that factorization of large integers is difficult
DSS/DSA	An encryption algorithm based on the discrete logarithm problem
Phrase	A passphrase used for protecting keys

Table 5.83.: Certificate/Key Terms

A single certificate can obtain the following ASN.1 attributes:

Attribute	Description
CN	The certificate owner's common name, mainly used to identify a host
C	The certificate owner's country (usually a TLD abbreviation)
ST	The certificate owner's state
L	The certificate owner's location
C	The certificate owner's country
O	The certificate owner's organization
OU	The name of the organizational unit to which the certificate issuer belongs
E	The certificate owner's email address

Table 5.84.: Certificate Attributes

Those attributes form a so-called subject name, mainly used for matching a certificate or when signing certificate requests:

```
Subject: C=CH, ST=Switzerland, L=Zurich, O=Company, OU=Networking,
         CN=router.company.com/Email=info@company.com
```

Depending on your configuration, keys and certificates may be used for particular services, for instance if OpenVPN uses a certificate-based authentication or if you want to access the system over HTTPS or SSH.

Keys and certificates can be installed to the system by uploading the corresponding files. It is also possible to create an own (unsigned) Certificate Authority and issue ready-for-use client certificates (e.g. for OpenVPN or WLAN clients). Such certificates can be revoked and invalidated again (for instance if they have been compromised or lost).

Generally, when generating keys and certificates on the box, the system's hostname is used for subject names and the passphrase corresponds to the current administrator's password.

Please note that an accurate system time is needed prior to creating certificates as it determines the lifetime of a certificate. The validity period is usually set to 10 years.

The X.509 certificates are encoded in PEM (Privacy Enhanced Mail) format and can be machined by OpenSSL (see www.openssl.org) or other Secure Sockets Layer tools. The .p12 files usually contain the CA certificate, the user certificate and the private key and the .phr files hold the required passphrase.

For curl-based SSL client connections as used by SDK functions or when downloading configuration/software images, you might upload the corresponding CA root certificates in order to build a chain of trust. Those can be derived from various browsers (see http://curl.haxx.se/docs/caextract.html). In case of uploaded CA bundles, all SSL client connections will abort if CA verification of the remote end fails.

5.8.7. Licensing

Certain features of NetModule routers require a valid license to be present in the system, some of them also depend on the mounted modules. Please contact us for getting a valid license for available components and we will provide a license file based on your serial number which can be installed to the router afterwards.

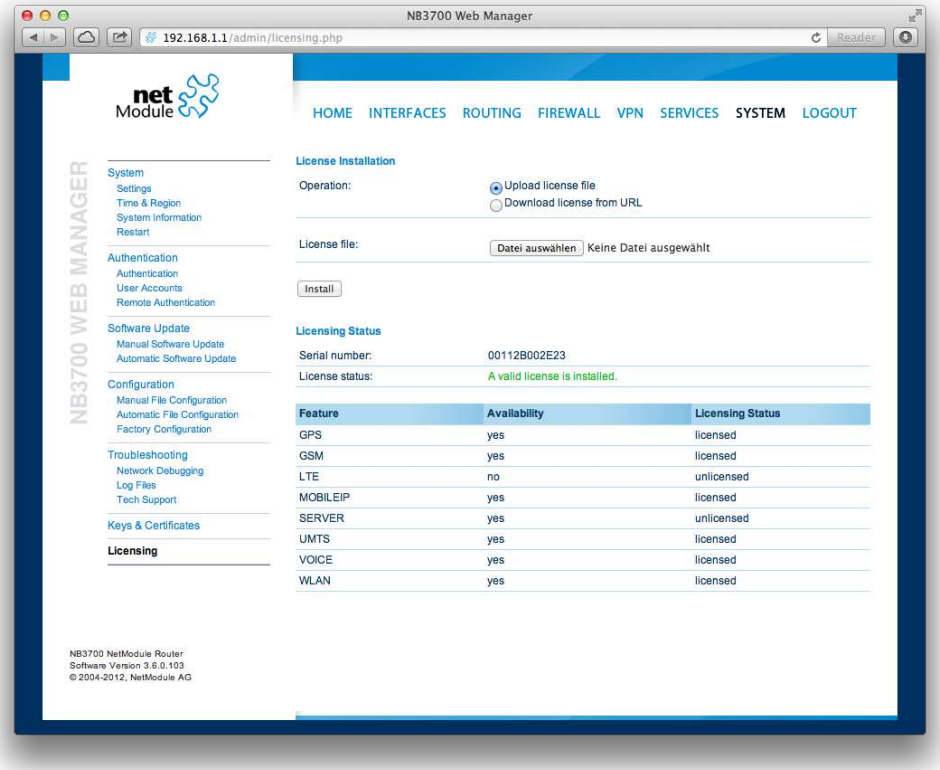


Figure 5.60.: Licensing

5.8.8. Legal Notice

OSS Notice

We inform you that NetModule products may contain in part open-source software. We are distributing such open-source software to you under the terms of GNU General Public License (GPL), GNU Lesser General Public License (LGPL) or other open-source licenses.

These licenses allow you to run, copy, distribute, study, change and improve any software covered by GPL, Lesser GPL, or other open-source licenses without any restrictions from us or our end user license agreement on what you may do with that software. Unless required by applicable law or agreed to in writing, software distributed under open-source licenses is distributed on an "AS IS" basis, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

To obtain the corresponding open source codes covered by these licenses, please contact our technical support at router@support.netmodule.com.

Acknowledgements

This product includes PHP, freely available from http://www.php.net.
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).
This product includes cryptographic software written by Eric Young(eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com).
This product includes software written Jean-loup Gailly and Mark Adler.
This product includes software MD5 Message-Digest Algorithm by RSA Data Security, Inc.
This product includes an implementation of the AES encryption algorithm based on code released by Dr Brian Gladman.

Copyright (C) 2014, NetModule. All rights reserved.

5.9. LOGOUT

Please use this menu to log out from Web Manager.

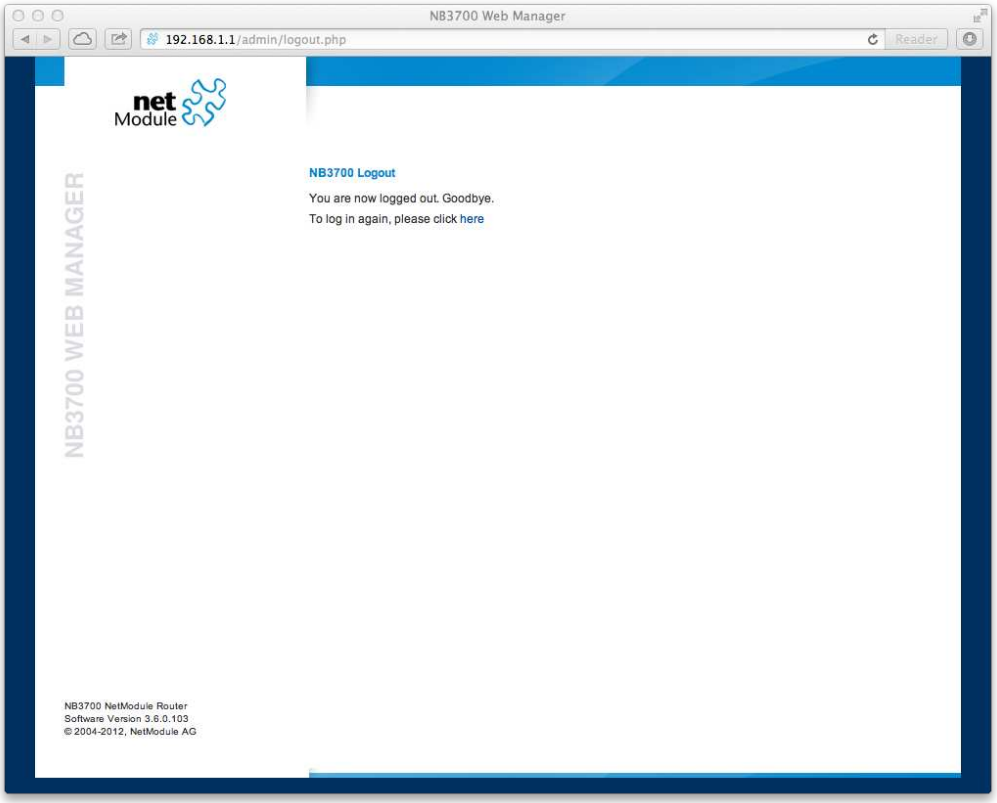


Figure 5.61.: Logout

6. Command Line Interface

The Command Line Interface (CLI) offers a generic control interface to the router and can be used to get/set configuration parameters, apply updates, restart services or perform other system tasks.

It will be started automatically in interactive mode when logging in as *admin* user or by running `cli -i`. However, the same syntax can be used when calling it from the system shell. A list of available commands can be displayed by running `cli -l`.

The CLI supports TAB completion, that is expanding entered words or fragments by hitting the TAB key at any time. This applies to commands but also to some arguments and generally offers a convenient way for working on the shell.

Please note that each CLI session will perform an automatic logout as soon as a certain time of inactivity (10 minutes by default) has been reached. It can be turned off by the command `no-autologout`.

6.1. General Usage

When operating the CLI in interactive mode, each entered command will be executed by the `RETURN` key. You can use the `Left` and `Right` keys to move the current point between entered characters or use the `Up` and `Down` keys to search the history of entered commands. Typing `exit` as well as pressing `CTRL-c` twice or `CTRL-d` on an empty command line will exit the CLI.

List of supported key sequences:

Key Sequence	Action
`CTRL-a`	Move to the start of the current line
`CTRL-e`	Move to the end of the line
`CTRL-f`	Move forward a character
`CTRL-b`	Move back a character
`ALT-f`	Move forward to the end of the next word
`ALT-b`	Move back to the start of the current or previous word

Key Sequence	Action
CTRL-l	Clear the screen leaving the current line at the top of the screen; with an argument given, refresh the current line without clearing the screen
CTRL-p	Fetch the previous command from the history list, moving back in the list
CTRL-n	Fetch the next command from the history list, moving forward in the list
ALT-<	Move to the first line in the history
ALT->	Move to the end of the input history
CTRL-r	Search backward starting at the current line and moving up through the history
CTRL-s	Freeze session
CTRL-q	Reactivate frozen session
CTRL-d	Delete character at point or exit CLI if at the beginning of the line
CTRL-t	Drag the character before point forward moving point forward as well; if point is at the end of the line, then this transposes the two characters before the point
ALT-t	Drag the word before point past the word after point, moving point over that word as well. If point is at the end of the line, this transposes the last two words on the line.
CTRL-k	Delete the text from point to the end of the line
CTRL-y	Yank the top of the deleted text into the buffer at point

Please note, that it can be required to apply quotes (") when entering commands with arguments containing whitespaces.
The following sections are now trying to explain the available commands.

6.2. Print Help

The `help` command can be used to get the list of available commands when called without arguments, otherwise it will print the usage of the specified command.

```
> help
```

```
Usage:
        help [<command>]

Available commands:

        get                     Get config parameters
        set                     Set config parameters
        update                  Update system facilities
        status                  Get status information
        scan                    Scan networks
        send                    Send message, mail or techsupport
        restart                 Restart service
        debug                   Debug system
        reset                   Reset system to factory defaults
        reboot                  Reboot system
        shell                   Run shell command
        help                    Print help for command
        no-autologout           Turn off auto-logout
        history                 Show command history
        exit                    Exit
```

6.3. Getting Config Parameters

The `get` command can be used to get configuration values.

```
> get -h
Usage:
        get [-hsvfc] <parameter> [<parameter>..]

Options:
        -s      generate sourceable output
        -v      validate config parameter
        -f      get factory default rather than current value
        -c      show configuration sections
```

6.4. Setting Config Parameters

The `set` command can be used to set configuration values.

```
> set -h
Usage:
```

```
        set [-hv] <parameter>=<value> [<parameter>=<value>..]

Options:
        -v      validate config parameter
```

6.5. Getting Status Information

The `status` command can be used to get various status information of the system.

```
> status -h
Usage:
        status [-hs] <section>

Options:
        -s      generate sourceable output

Available sections:

        summary             Short status summary
        info                System and config information
        config              Current configuration
        system              System information
        configuration       Configuration information
        license             License information
        wwan                WWAN module status
        wlan                WLAN module status
        gnss                GNSS (GPS) module status
        eth                 Ethernet interface status
        lan                 LAN interface status
        wan                 WAN interface status
        openvpn             OpenVPN connection status
        ipsec               IPsec connection status
        pptp                PPTP connection status
        gre                 GRE connection status
        dialin              Dial-In connection status
        mobileip            MobileIP status
        dio                 Digital IO status
        sms                 SMS status
        firewall            Firewall status
        qos                 QoS status
        neigh               Neighborhood status
        location            Current Location
```

6.6. Scanning Networks

The scan command can be used to scan for available WWAN and WLAN networks.

```
> scan -h
Usage:
        scan [-hs] <interface>

Options:
        -s      generate sourceable output
```

6.7. Sending E-Mail or SMS

The send command can be used to send a message via E-Mail/SMS to the specified address or phone number.

```
> send -h
Name:
        cli-send (Send message or mail)

Usage:
        send [-h] <type> <dest> <msg>

Options:
        <type>      type of message to be sent (mail, sms,
        techsupport)
        <dest>      destination of message (mail-address or phone-
        number)
        <msg>       message to be sent
```

6.8. Updating System Facilities

The update command can be used to perform various system updates.

```
> update -h
Usage:
        update [-hrsn] <software|config|license|sshkeys> <URL>

Options:
        -r      reboot after update
```

```
        -n      don't reset missing config values with factory
        defaults
        -s      show update status


Available actions:

        software            Perform software update
        config              Update configuration
        license             Update licenses
        sshkeys             Install SSH authorized keys

You may also run 'update software latest' to install the latest
version
from our server.
```

6.9. Restarting Services

The `restart` command can be used to restart system services.

```
> restart -h
Usage:
        restart [-h] <service>

Available services:

        link-manager          WAN links
        wwan-manager          WWAN manager
        wlan                  WLAN interfaces
        network               Networking
        dnsmasq               DNS/DHCP server
        configd               Configuration daemon
        firewall              Firewall and NAPT
        lighttpd              HTTP server
        openvpn               OpenVPN connections
        gre                   GRE connections
        ipsec                 IPsec connections
        pptp                  PPTP connections
        snmpd                 SNMP daemon
        syslog                Syslog daemon
        telnet                Telnet server
        dropbear              SSH server
        vrrpd                 VRRP daemon
```

```
        usbipd                  USB/IP daemon
        surveyor                Supervision daemon
        voiced                  Voice daemon
        gpsd                    GPS daemon
        smsd                    SMS daemon
```

6.10. Debug System

The `debug` command can be used to obtain debug/log messages.

```
> debug -h
Usage:
        debug [-h] <target>

Available debug targets:

        system
        scripts
        configd
        watchdog
        swupdate
        wwan-manager
        led-manager
        event-manager
        link-manager
        wwanmd
        surveyor
        mobile-node
        home-agent
        voiced
        smsd
        sdkhost
        qmid
        ser2net
        qosd
```

6.11. Resetting System

The `reset` command can be used to reset the router back to factory defaults.

```
> reset -h
```

```
Usage:
        reset [-h]
```

6.12. Rebooting System

The `reboot` command can be used to reboot the router.

```
> reboot -h
Usage:
        reboot [-h]
```

6.13. Running Shell Commands

The `shell` command can be used to execute a system shell and run any arbitrary application or script.

```
> shell -h
Usage:
        shell [-h] [<cmd>]
```

6.14. Working with History

The `history` command will print the list of entered commands on a per-user basis.

```
> histor -h
Usage:
        history [-c]
```

It can be cleared by `history -c`.

6.15. CLI-PHP

CLI-PHP, the HTTP frontend to the CLI application, can be used to configure and control the router remotely. It is enabled in factory configuration, thus can be used for deployment purposes, but disabled as soon as the admin account has been set up.

The service can later be turned on/off by setting the `cliphp.status` configuration parameter:

```
    cliphp.status=0         Service is disabled
```

```
    cliphp.status=1          Service is enabled
```

This section describes the CLI-PHP interface for Version 2. It accepts POST and GET requests.
Running with GET requests, the general usage is defined as follows:

```
Usage:
  http(s)://cli.php?<key1>=<value1>&<key2>=<value2>..<keyN>=<valueN>

      Available keys:

      output           Output format (html, plain)
      usr              Username to be used for authentication
      pwd              Password to be used for authentication
      command          Command to be executed
      arg0..arg31      Arguments passed to commands

Notes:
  The commands correspond to CLI commands as seen by 'cli -l', the
  arguments (arg0..arg31) will be directly passed to cli.

  Thus, an URL containing the following sequence:

   command=get&arg0=admin.password&arg1=admin.debug&arg2=admin.
   access

  will lead to cli being called as:

   cli get "admin.password" "admin.debug" "admin.access"

  It supports whitespaces but please be aware that any special
  characters in
  the URL must be specified according to RFC1738 (usually done by
  common
  clients such as wget, lynx, curl).

Response:
  The returned response will always contain a status line in the
  format:

  <return>: <msg>

  with return values of OK if succeeded and ERROR if failed. Any
  output from the commands will be appended.
```

```
Examples:
  OK: status command successful
  ERROR: authentication failed
```

status - Display status information

```
Key usage:
  command=status[&arg0=<section>]

Notes:
  Available sections can be retrieved by running
  command=status&arg0=-h.
  Please note that the status summary can be displayed without
  authentication.

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=status&arg0=-h

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=status&arg0=summary

  http://192.168.1.1/cli.php?version=2&output=html&command=status
```

get - Get configuration parameter

```
Key usage:
  command=get&arg0=<config-key>[&arg1=<config-key>..]

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=get&arg0=config.version

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=get&arg0=openvpn.status&arg1=snmp.status&arg2=
  ipsec.status
```

set - Set configuration parameter

```
Key usage:
  command=set&arg0=<config-key>&arg1=<config-value>[&arg2=<config-
```

```
  key>&arg3=<config-value>..]

Notes:
  In contrast to the other commands, this command requires a set of
  tuples because of the reserved '=' char, i.e.
  [arg0=key0, arg1=val0], [arg2=key1, arg3=val1], [arg4=key2, arg5=
  val2], etc

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=set&arg0=snmp.status&arg1=1

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=set&arg0=snmp.status&arg1=0&arg2=openvpn.status&
  arg3=1
```

restart - Restart a system service

```
Key usage:
  command=restart&arg0=<service>

Notes:
  Available services can be retrieved by running 'command=restart&
  arg0=-h'

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=restart&arg0=-h

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=restart&arg0=link-manager
```

reboot - Trigger system reboot

```
Key usage:
  command=reboot

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=reboot
```

reset - Run factory reset

```
Key usage:
  command=reset

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=reset
```

update - Update system facilities

```
Key usage:
  command=update&arg0=<facility>&arg1=<URL>

Notes:
  Available facilities can be retrieved by running 'command=update&
  arg0=-h'

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=update&arg0=software&arg1=tftp://192.168.1.254/
  latest

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=update&arg0=config&arg1=tftp://192.168.1.254/user-
  config.zip

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=update&arg0=license&arg1=http://192.168.1.254/xxx.
  lic
```

7. Technical Support

NetModule's mission statement is to provide you with state of the art products, technologies and services for your embedded applications. This certainly includes a professional and friendly team of support engineers which will be pleased to offer consultancy, provide assistance and deliver solutions in case of technical issues. With their broad-based experience they will be able to narrow down your problem and thus prevent you from getting too much gray hair.

In case of support requests please use our support form on the NetModule web page and submit a detailed description of your problem together with a tech-support file which contains all the necessary information to speed up the process of analyzing and resolving your problem.

The latest software and documentation material can found in the technical support area via the NetModule website.

Feedback

Your feedback is highly appreciated; please send comments, suggestions, feature requests, error reports or your personal user experience with this NB3700 router to router@support.netmodule.com.

8. Legal Notice

Copyright

This document contains proprietary information of NetModule. No parts of the work described herein may be reproduced. Reverse engineering of the hardware or software is prohibited and protected by patent law. This material or any portion of it may not be copied in any form or by any means, stored in a retrieval system, adopted or transmitted in any form or by any means (electronic, mechanical, photographic, graphic, optic or otherwise), or translated in any language or computer language without the prior written permission of NetModule.

The information in this document is subject to change without notice. We would like to point out that NetModule makes no representation or warranties with respect to the contents herein and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this information.

This document may contain information about third party products or processes. Such third party information is generally out of influence of NetModule and therefore NetModule shall not be responsible for the correctness or legitimacy of this information. If you experience any incorrect or erroneous specifications in the documentation, please report them in writing by email to router@support.netmodule.com. While due care has been taken to deliver accurate documentation, NetModule does not warrant that this document is error-free.

NetModule and *NB3700* are trademarks and the logo is a service mark of NetModule AG, Switzerland.

All other products or company names mentioned herein are used for identification purposes only and may be trademarks or registered trademarks of their respective owners. The following description of software, hardware or process of NetModule or other third party provider may be included with your product and will be subject to the software, hardware or other license agreements.

Contact

Please contact us for up-to-date product descriptions, documentation, application notes, firmware upgrades, troubleshooting tips, press releases or any other concerns.

NetModule AG
Meriedweg 11
CH-3172 Niederwangen
Switzerland

Tel +41 31 985 25 10
Fax +41 31 985 25 11
info@netmodule.com
http://www.netmodule.com

Copyright ©2014 NetModule AG, Switzerland All rights reserved

A. Appendix

A.1. Abbrevations

Parameter	Description
ETHx	Corresponds to Ethernet interfaces (either single or switched ones)
LANx	LAN interfaces which are generally based on Ethernet interfaces (including bridges)
WLANx	Refers to a Wireless LAN interface which will be represented as additional LAN interface when configured as access point
WWANx	Refers to a Wireless Wide Area Network (2G/3G/4G) connection
TUNx	Specifies an OpenVPN tunnel interface (based on TUN)
TAPx	Specifies an OpenVPN tunnel interface (based on TAP)
PPTPx	Specifies a PPTP tunnel interface
MOBILEIPx	Refers to a Mobile IP tunnel interface
SIMx	Specifies the SIM slot as seen on the front panel
GNSSx	Specifies a Global Navigation Satellite System module
Mobilex	Identifies a WWAN modem
SERIALx	Identifies a serial port
OUTx	Specifies a digital I/O output port (DOx)
INx	Specifies a digital I/O input port (DIx)
ANY	Generally includes all options offered by the current section
APN	Access Point Name
CID	A Cell ID is a generally unique number used to identify each Base Transceiver Station (BTS).

Parameter	Description
LAC	The Location Area Code corresponds to an identifier of a set of base stations that are grouped together to optimize signaling
LAI	The Location Area Identity is a globally unique number that identifies the country, network provider and location area
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
DNS	Domain Name System
NAPT	Network Address and Port Translation
DHCP	Dynamic Host Configuration Protocol
SDK	Script Development Kit which can be used to program applications
CLI	Command Line Interface, a generic interface to query the router or perform system tasks
SIM	Subscriber Identity Module
SMS	Short Message Service
SSID	Service Set Identifiers, can be used to define multiple WLAN networks on a module
STP	Spanning Tree Protocol
USSD	Unstructured Supplementary Service Data
VRRP	Virtual Router Redundancy Protocol
VPN	Virtual Private Network
WAN	WAN links include all Wide Area Network interfaces which are currently activated in the system
FQDN	Fully qualified domain name

Table A.1.: Abbreviations

In general, internal interfaces are written lower-case and may have a different naming. Their index starts from zero, whereas interfaces seen by the user will be written in capital letters starting from one.

A.2. System Events

ID	Event	Description
101	wan-up	WAN link came up
102	wan-down	WAN link went down
201	dio-in1-on	DIO IN1 turned on
202	dio-in1-off	DIO IN1 turned off
203	dio-in2-on	DIO IN2 turned on
204	dio-in2-off	DIO IN2 turned off
205	dio-out1-on	DIO OUT1 turned on
206	dio-out1-off	DIO OUT1 turned off
207	dio-out2-on	DIO OUT2 turned on
208	dio-out2-off	DIO OUT2 turned off
301	gps-up	GPS signal is available
302	gps-down	GPS signal is not available
401	openvpn-up	OpenVPN connection came up
402	openvpn-down	OpenVPN connection went down
403	ipsec-up	IPsec connection came up
404	ipsec-down	IPsec connection went down
406	pptp-up	PPTP connection came up
407	pptp-down	PPTP connection went down
408	dialin-up	Dial-In connection came up
409	dialin-down	Dial-In connection went down
410	mobileip-up	Mobile IP connection came up
411	mobileip-down	Mobile IP connection went down
412	gre-up	GRE connection came up
413	gre-down	GRE connection went down
501	system-login-failed	User login failed
502	system-login-succeeded	User login succeeded

ID	Event	Description
503	system-logout	User logged out
504	system-rebooting	System reboot has been triggered
505	system-startup	System has been started
506	test	test event
507	sdk-startup	SDK has been started
508	system-time-updated	System time has been updated
601	sms-sent	SMS has been sent
602	sms-notsent	SMS has not been sent
603	sms-received	SMS has been received
604	sms-report-received	SMS report has been received
701	call-incoming	A voice call is coming in
702	call-outgoing	Outgoing voice call is being established
801	ddns-update-succeeded	Dynamic DNS update succeeded
802	ddns-update-failed	Dynamic DNS update failed
901	usb-storage-added	USB storage device has been added
902	usb-storage-removed	USB storage device has been removed
903	usb-eth-added	USB Ethernet device has been added
904	usb-eth-removed	USB Ethernet device has been removed
905	usb-serial-added	USB serial device has been added
906	usb-serial-removed	USB serial device has been removed

Table A.2.: System Events

A.3. Factory Configuration

The factory configuration including default values for any configuration parameter can be derived from the file `/etc/config/factory-config.cfg` on the router. You may also call `cli get -f <parameter>` for obtaining a specific default value.

A.4. SNMP VENDOR MIB

```
-- ************************************************************************
-- NetModule AG VENDOR MIB
--
--
-- (c) COPYRIGHT 2014 by NetModule AG, Switzerland
-- All rights reserved.
--
--
-- ************************************************************************

NB-MIB DEFINITIONS ::= BEGIN

-- ************************************************************************
-- imports
-- ************************************************************************

IMPORTS
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
       Integer32, Counter32, Gauge32,
       Counter64, TimeTicks                    FROM SNMPv2-SMI
       TEXTUAL-CONVENTION, DisplayString,
       PhysAddress, TruthValue, RowStatus,
       TimeStamp, AutonomousType, TestAndIncr  FROM SNMPv2-TC
       MODULE-COMPLIANCE, OBJECT-GROUP         FROM SNMPv2-CONF
       snmpTraps                               FROM SNMPv2-MIB
       URLString                               FROM NETWORK-SERVICES-MIB
       enterprises                             FROM SNMPv2-SMI;

-- ************************************************************************
-- module definition
-- ************************************************************************

nb MODULE-IDENTITY
        LAST-UPDATED "201405091000Z"
        ORGANIZATION "NetModule AG"
        CONTACT-INFO
            "NetModule AG, Switzerland"
        DESCRIPTION
            "MIB module which defines the NB router specific entities"

        REVISION "201405091000Z"
        DESCRIPTION
             "MIB for software release 3.7"

        REVISION "201212191000Z"
        DESCRIPTION
            "MIB for software release 3.6"
        ::= { netmodule 10 }


-- ************************************************************************
-- root anchor
-- ************************************************************************

netmodule OBJECT IDENTIFIER ::= { enterprises 31496 }


-- ************************************************************************
-- table definitions
-- ************************************************************************

system        OBJECT IDENTIFIER ::= { nb   1 }
products      OBJECT IDENTIFIER ::= { nb  10 }
admin         OBJECT IDENTIFIER ::= { nb  40 }
dio           OBJECT IDENTIFIER ::= { nb  53 }
traps         OBJECT IDENTIFIER ::= { nb 100 }

-- ************************************************************************

nb1600        OBJECT IDENTIFIER ::= { products 46 }
nb2700        OBJECT IDENTIFIER ::= { products 47 }
nb3700        OBJECT IDENTIFIER ::= { products 48 }

-- ************************************************************************
-- NBAdminTable
-- ************************************************************************

swVersion OBJECT-TYPE
    SYNTAX      DisplayString
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The currently installed system software version"
    ::= { admin 1 }

kernelVersion OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The currently installed kernel version"
    ::= { admin 2 }

serialNumber OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The serial number of the device"
    ::= { admin 3 }

deviceRestart OBJECT-TYPE
    SYNTAX  INTEGER {
                restart (1)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Force a device restart"
    ::= { admin 10 }

configUpdate OBJECT-TYPE
    SYNTAX  URLString
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Update the system configuration from the specified URL.
          The URL must be preceded by one of the prefixes tftp://, ftp://, http://
          and either point to the update package or to a server directory which
          contains a file named <serial-number>.zip"
    ::= { admin 11 }

configUpdateStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                succeeded (1),
                failed (2),
                inprogress (3),
                notstarted (4)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The status of the last configuration update cycle"
    ::= { admin 12 }

softwareUpdate OBJECT-TYPE
    SYNTAX  URLString
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Update the system software from the specified URL,
          the URL must be preceded by one of the prefixes tftp://, ftp://, http://
          and point to the to be installed image."
    ::= { admin 13 }

softwareUpdateStatus OBJECT-TYPE
    SYNTAX  INTEGER {
                succeeded (1),
                failed (2),
                inprogress (3),
                notstarted (4)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The status of the last software update cycle"
    ::= { admin 14 }


-- ****************************************************************************
-- NBWwanTable
-- ****************************************************************************

nbWwanTable OBJECT-TYPE
```

```
    SYNTAX      SEQUENCE OF NBWwanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The table describing any WWAN modems and their current settings"
    ::= { nb 50 }

nbWwanEntry OBJECT-TYPE
    SYNTAX      NBWwanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An entry describing a WWAN modem and its current settings"
    INDEX       { wwanModemIndex }
    ::= { nbWwanTable 1 }

NBWwanEntry ::= SEQUENCE {
    wwanModemIndex Integer32,
    wwanModemName DisplayString,
    wwanModemType DisplayString,
    wwanServiceType DisplayString,
    wwanRegistrationState DisplayString,
    wwanSignalStrength Integer32,
    wwanNetworkName DisplayString,
    wwanLocalAreaIdentification DisplayString,
    wwanLocalAreaCode DisplayString,
    wwanCellId DisplayString
}

wwanModemIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
    "WWAN modem index"
    ::= { nbWwanEntry 1 }

wwanModemName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WWAN modem name"
    ::= { nbWwanEntry 2 }

wwanModemType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WWAN modem type"
    ::= { nbWwanEntry 3 }

wwanServiceType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current service type of the WWAN modem"
    ::= { nbWwanEntry 4 }

wwanRegistrationState OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current registration state of the WWAN modem"
    ::= { nbWwanEntry 5 }

wwanSignalStrength OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current signal strength of the WWAN modem (-999 means unknown)"
    ::= { nbWwanEntry 6 }

wwanNetworkName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The network name to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 7 }

wwanLocalAreaIdentification OBJECT-TYPE
    SYNTAX      DisplayString
```

```
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Local Area Identification (LAI) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 8 }

wwanLocalAreaCode OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Local Area Code (LAC) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 9 }

wwanCellId OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Cell ID (CID) to which the WWAN modem is currently registered"
    ::= { nbWwanEntry 10 }


-- ****************************************************************************
-- NBGnssTable
-- ****************************************************************************

nbGnssTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NBGnssEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The table describing any GNSS devices and their current settings"
    ::= { nb 51 }

nbGnssEntry OBJECT-TYPE
    SYNTAX      NBGnssEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry describing a GNSS device and its current settings"
    INDEX       { gnssIndex }
    ::= { nbGnssTable 1 }

NBGnssEntry ::= SEQUENCE {
    gnssIndex Integer32,
    gnssName DisplayString,
    gnssSystem DisplayString,
    gnssLat DisplayString,
    gnssLon DisplayString,
    gnssAlt DisplayString,
    gnssNumSat Integer32
}

gnssIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "GNSS device index"
    ::= { nbGnssEntry 1 }

gnssName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "GNSS device name"
    ::= { nbGnssEntry 2 }

gnssSystem OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "GNSS system used by the device"
    ::= { nbGnssEntry 3 }

gnssLat OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current latitude value received by the GNSS device"
    ::= { nbGnssEntry 4 }
```

```
gnssLon OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current longitude value received by the GNSS device"
    ::= { nbGnssEntry 5 }

gnssAlt OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current altitude value received by the GNSS device"
    ::= { nbGnssEntry 6 }

gnssNumSat OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The current number of available satellites for the GNSS device"
    ::= { nbGnssEntry 7 }


-- ****************************************************************************
-- NBWlanTable
-- ****************************************************************************

nbWlanTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF NBWlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table describing any WLAN modems and their current settings."
    ::= { nb 60 }

nbWlanEntry OBJECT-TYPE
    SYNTAX      NBWlanEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry describing a WLAN modem and its current settings."
    INDEX       { wlanModuleIndex }
    ::= { nbWlanTable 1 }

NBWlanEntry ::= SEQUENCE {
    wlanModuleIndex Integer32,
    wlanModuleName DisplayString,
    wlanModuleType DisplayString,
    wlanNumClients Integer32

}

wlanModuleIndex OBJECT-TYPE
    SYNTAX      Integer32(0..254)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "WLAN module index"
    ::= { nbWlanEntry 1 }

wlanModuleName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WLAN module name"
    ::= { nbWlanEntry 2 }

wlanModuleType OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "WLAN module type"
    ::= { nbWlanEntry 3 }

wlanNumClients OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Current number of clients connected to the WLAN module (if operated as access point)"
```

```
    ::= { nbWlanEntry 4 }


-- ****************************************************************************
-- NBDioTable
-- ****************************************************************************

dioStatusIn1 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The current value of digital I/O port IN1"
    ::= { dio 1 }

dioStatusIn2 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The current value of digital I/O port IN2"
    ::= { dio 2 }

dioStatusOut1 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The current value of digital I/O port OUT1"
    ::= { dio 3 }

dioStatusOut2 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The current value of digital I/O port OUT2"
    ::= { dio 4 }

dioSetOUT1 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "The update value for digital I/O port OUT1"
    ::= { dio 10 }

dioSetOUT2 OBJECT-TYPE
    SYNTAX  INTEGER {
                off (0),
                on (1)
            }
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "The update value for digital I/O port OUT2"
    ::= { dio 11 }


-- ****************************************************************************
-- trap objects
-- ****************************************************************************

events          OBJECT IDENTIFIER ::= { traps 0 }

wan-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "WAN link came up"
    ::= { events 101 }
```

```
wan-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "WAN link went down"
    ::= { events 102 }

dio-in1-on NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO IN1 turned on"
    ::= { events 201 }

dio-in1-off NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO IN1 turned off"
    ::= { events 202 }

dio-in2-on NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO IN2 turned on"
    ::= { events 203 }

dio-in2-off NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO IN2 turned off"
    ::= { events 204 }

dio-out1-on NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO OUT1 turned on"
    ::= { events 205 }

dio-out1-off NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO OUT1 turned off"
    ::= { events 206 }

dio-out2-on NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO OUT2 turned on"
    ::= { events 207 }

dio-out2-off NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "DIO OUT2 turned off"
    ::= { events 208 }

gps-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "GPS signal is available"
    ::= { events 301 }

gps-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "GPS signal is not available"
    ::= { events 302 }

openvpn-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "OpenVPN connection came up"
    ::= { events 401 }

openvpn-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "OpenVPN connection went down"
    ::= { events 402 }

ipsec-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "IPsec connection came up"
    ::= { events 403 }

ipsec-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "IPsec connection went down"
    ::= { events 404 }

pptp-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "PPTP connection came up"
    ::= { events 406 }

pptp-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "PPTP connection went down"
    ::= { events 407 }
```

```
dialin-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Dial-In connection came up"
    ::= { events 408 }

dialin-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Dial-In connection went down"
    ::= { events 409 }

mobileip-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Mobile IP connection came up"
    ::= { events 410 }

mobileip-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Mobile IP connection went down"
    ::= { events 411 }

gre-up NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "GRE connection came up"
    ::= { events 412 }

gre-down NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "GRE connection went down"
    ::= { events 413 }

system-login-failed NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "User login failed"
    ::= { events 501 }

system-login-succeeded NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "User login succeeded"
    ::= { events 502 }

system-logout NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "User logged out"
    ::= { events 503 }

system-rebooting NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "System reboot has been triggered"
    ::= { events 504 }

system-startup NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "System has been started"
    ::= { events 505 }

test NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "test event"
    ::= { events 506 }

sdk-startup NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "SDK has been started"
    ::= { events 507 }

system-time-updated NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "System time has been updated"
    ::= { events 508 }

sms-sent NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "SMS has been sent"
    ::= { events 601 }

sms-notsent NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "SMS has not been sent"
    ::= { events 602 }

sms-received NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "SMS has been received"
    ::= { events 603 }
```

```
sms-report-received NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "SMS report has been received"
    ::= { events 604 }

call-incoming NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "A voice call is coming in"
    ::= { events 701 }

call-outgoing NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Outgoing voice call is being established"
    ::= { events 702 }

ddns-update-succeeded NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Dynamic DNS update succeeded"
    ::= { events 801 }

ddns-update-failed NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "Dynamic DNS update failed"
    ::= { events 802 }

usb-storage-added NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "USB storage device has been added"
    ::= { events 901 }

usb-storage-removed NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "USB storage device has been removed"
    ::= { events 902 }

usb-eth-added NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "USB Ethernet device has been added"
    ::= { events 903 }

usb-eth-removed NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "USB Ethernet device has been removed"
    ::= { events 904 }

usb-serial-added NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "USB serial device has been added"
    ::= { events 905 }

usb-serial-removed NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION "USB serial device has been removed"
    ::= { events 906 }

END
```

A.5. SDK Examples

Event	Description
config-summary.are	This script shows a summary of the currently running configuration.
dio-monitor.are	This script monitors the DIO ports and sends a SMS to the specified phone number.
dio-server.are	This script implements a TCP server which can be used to control the DIO ports.
dio.are	This script can be used to set a digital output port.
dynamic-operator.are	This script will scan Mobile2 and dial the appropriate SIM on Mobile1
email-to-sms.are	This script implements a lightweight SMTP server which is able to receive mail and forward them as SMS to a phone number.
etherwake.are	This script can be used to wake up a sleeping host (WakeOnLan)
gps-monitor.are	A script for activating WLAN as soon as GPS position (lat,lon) is within a specified range.
gps-udp-client.are	This script sends the local GPS NMEA stream to a remote UDP server.
gps-upd-client-compat.are	This script sends the local GPS NMEA stream to a remote UDP server (incl. device identity).
led.are	This script can be used to set a LED
mount-media.are	This script can be used to mount an USB storage stick.
ping-supervision.are	This script will supervise a specified host.
read-config.are	This script can be used to read a configuration parameter.
scan-mobile.are	This script can be used to switch the Mobile LAI according to available networks
scan-wlan.are	This script can be used to switch the WLAN client network according to availability
send-mail.are	This script will send an E-Mail to the specified address.
send-sms.are	This script will send an SMS to the specified phone number.

Event	Description
serial-read.are	This script can be used to read messages from the serial port.
serial-readwrite.are	This script will write to and read from the serial port.
serial-tcsetattr.are	This script can be used to set/get the attributes of the serial port.
serial-udp-server.are	This script reads messages coming from the serial port and forwards them via UDP to a remote host (and vice versa).
serial-write.are	This script can be used to write a message to the serial port.
sms-control.are	This script will execute commands received by SMS.
sms-delete-inbox.are	This script can be used to flush the SMS inbox.
sms-read-inbox.are	This script can be used to read the SMS inbox.
sms-to-email.are	This script will forward incoming SMS messages to a given E-mail address.
sms-to-serial.are	This script can be used to write a received SMS to the serial port.
snmp.are	This script can be used to send SNMP traps
status.are	This script can be used to display all status variables
syslog.are	Throw a simple syslog message.
tcpclient.are	This script sends a message to a TCP server.
tcpserver.are	This script implements a TCP server which is able to receive messages.
transfer.are	This scripts stores the latest GNSS positions in a remote FTP file
udp-msg-server.are	This script will run an UDP server which is able to receive messages and forward them as SMS/E-Mail.
udpclient.are	This script sends a message to a remote UDP server.
udpserver.are	This script implements an UDP server which is able to receive messages.
update-config.are	This script can be used to perform a configuration update
webpage.are	This script will generate a page which can be viewed in the Web Manager

Event	Description
write-config.are	This script can be used to set a configuration parameter.

Table A.3.: SDK Examples

SDK API Manual

Version 1.3

NetModule AG, Switzerland

October 29, 2014

Contents

1 Introduction **3**

2 API functions **4**

2.1 Serial . . . 4
2.1.1 nb_serial_getattr . . . 4
2.1.2 nb_serial_setattr . . . 4
2.1.3 nb_serial_write . . . 4
2.1.4 nb_serial_read . . . 5
2.2 Media . . . 5
2.2.1 nb_media_mount . . . 5
2.2.2 nb_media_umount . . . 5
2.2.3 nb_media_getmount . . . 5
2.3 SMS . . . 6
2.3.1 nb_sms_send . . . 6
2.3.2 nb_sms_list . . . 6
2.3.3 nb_sms_retrieve . . . 6
2.3.4 nb_sms_header . . . 6
2.3.5 nb_sms_body . . . 7
2.3.6 nb_sms_delete . . . 7
2.4 E-mail . . . 7
2.4.1 nb_email_send . . . 7
2.5 Digital I/O . . . 8
2.5.1 nb_dio_get . . . 8
2.5.2 nb_dio_set . . . 8
2.5.3 nb_dio_summary . . . 8
2.6 Configuration . . . 8
2.6.1 nb_config_get . . . 8
2.6.2 nb_config_set . . . 9
2.6.3 nb_config_summary . . . 9
2.7 Status Information . . . 9
2.7.1 nb_status . . . 9
2.7.2 nb_status_summary . . . 10
2.8 Network Scanning . . . 10

2.8.1 nb_scan_networks . . . 10
2.9 File Transfers . . . 10
2.9.1 nb_transfer_get . . . 10
2.9.2 nb_transfer_put . . . 11
2.9.3 nb_transfer_post . . . 11
2.9.4 nb_transfer_list . . . 12
2.9.5 nb_transfer_delete . . . 12
2.10 LED . . . 12
2.10.1 nb_led_acquire . . . 12
2.10.2 nb_led_release . . . 13
2.10.3 nb_led_set . . . 13
2.11 Config/Software Update . . . 13
2.11.1 nb_update_status . . . 14
2.11.2 nb_update_config . . . 14
2.11.3 nb_update_software . . . 14
2.11.4 nb_update_sshkeys . . . 14
2.12 Web Pages . . . 15
2.12.1 nb_page_register . . . 15
2.12.2 nb_page_unregister . . . 15
2.12.3 nb_page_request . . . 15
2.12.4 nb_page_respond . . . 16
2.12.5 nb_page_finish . . . 16
2.13 SNMP functions . . . 16
2.13.1 nb_snmp_traphost . . . 16
2.13.2 nb_snmp_trap . . . 17
2.14 Various network-related functions . . . 17
2.14.1 nb_gethostbyname . . . 17
2.14.2 nb_ifc_address . . . 18
2.14.3 nb_ping . . . 18
2.14.4 nb_arp_ping . . . 18
2.14.5 nb_arp_gratuitous . . . 18
2.14.6 nb_etherwake . . . 19
2.15 Other system functions . . . 19
2.15.1 nb_syslog . . . 19
2.15.2 nb_event_get . . . 19
2.15.3 nb_reboot . . . 19

1 Introduction

This manual describes the SDK API functions which provide a range of general-purpose extensions for the Arena scripting language.

Version 1.2 ships with the following features:

```
- Send and retrieve SMS
- Send E-mail
- Read or write from or to a serial device
- Control digital input/output ports
- Run TCP/UDP servers
- Run IP/TCP/UDP clients
- Access files of mounted media (e.g. an USB stick)
- Retrieve status information from the system
- Get or set configuration values
- Write to syslog
- Transfer files over HTTP/FTP
- Perform config/software updates
- Control LEDs
- Get system events or reboot system
- Scan available networks
- Web Pages
- SNMP functions
- Various network-related functions
- Other system-related functions
```

2 API functions

2.1 Serial

2.1.1 nb_serial_getattr

```
struct nb_serial_getattr (string dev)
```

The nb_serial_getattr function retrieves the current attributes associated to a serial device.

```
dev         serial device (e.g. serial0 for first device)
```

Returns a struct containing values for baudrate, databit, stopbit, parity, flowctl or void on error.

2.1.2 nb_serial_setattr

```
int nb_serial_setattr (string dev, int b, int d, int s, int p, int
    f)
```

The nb_serial_setattr function can be used to set the attributes of a serial device.

```
dev         serial device (e.g. serial0 for first device)
b           baudrate e.g. 9600, 19200, 38400, 57600, 115200
d           number of data bits (5, 6, 7, 8)
s           number of stop bits (1, 2)
p           parity (0=no parity, 1=odd parity, 2=even parity)
f           flow control (0=none, 1=xon/xoff, 2=hardware)
```

Returns -1 on error otherwise zero.

2.1.3 nb_serial_write

```
int nb_serial_write (string dev, string msg)
```

The nb_serial_write function can be used to directly write a message to a serial device.

```
dev        serial device (e.g. serial0 for first device)
msg        message to be written
```

Returns number of bytes write or -1 on error.

2.1.4 nb_serial_read

```
string nb_serial_read (string dev)
```

The nb_serial_read function can be used to read a message from a serial device.

```
dev        serial device (e.g. serial0 for first device)
```

Returns the string received from the serial port or an empty string in case of errors.

2.2 Media

2.2.1 nb_media_mount

```
int nb_media_mount (string dev)
```

The nb_media_mount function mounts the specified media device.

```
dev        device name (eg. usb0)
```

Returns 0 on success and -1 on error. The media will be mounted in e.g. /mnt/media/usb0, you can use any IO functions to operate on it.

2.2.2 nb_media_umount

```
int nb_media_umount (string dev)
```

The nb_media_umount function umounts the specified media device.

```
dev        device name (e.g usb0)
```

Returns -1 on error.

2.2.3 nb_media_getmount

```
string nb_media_getmount (void)
```

The nb_media_getmount function returns all media currently mounted including the corresponding mountpoint. Returns list of mounted media and its mountpoints line-by-line (such as "xx on xx/xx"). If nothing is mounted or in case of an error an empty string will be returned.

2.3 SMS

Please note that the SMS daemon must be properly configured prior to using the functions.

2.3.1 nb_sms_send

```
string nb_sms_send (string phonenum, string msg)
```

The nb_sms_send function can be used to send a SMS to the specified telephone number.

```
phonenum    recipient's phone number (international format)
msg         message to be sent
```

Returns message id on success or NULL on error.

2.3.2 nb_sms_list

```
array nb_sms_list (void)
```

The nb_sms_list function can be used to retrieve the messages in the inbox. Returns an array of message files.

2.3.3 nb_sms_retrieve

```
string nb_sms_retrieve (string file)
```

The nb_sms_retrieve function returns the whole message specified by file.

2.3.4 nb_sms_header

```
string nb_sms_header (string file, string tag)
```

The nb_sms_header function returns the headers of a given message.

```
file    the message file
tag     a specific header tag (such as "From")
```

Returns header value of specified tag, all headers if omitted or NULL on error.

2.3.5 nb_sms_body

```
string nb_sms_body (string file)
```

The nb_sms_body function returns the body of a given message.

```
file    the message file
```

Returns the message's body text or NULL on error.

2.3.6 nb_sms_delete

```
int nb_sms_delete (string file)
```

The nb_sms_delete function can be used to delete a message from the inbox.

```
file    the message file
```

Returns 0 on success or -1 on error.

2.4 E-mail

Please note that the E-Mail client must be properly configured prior to using the functions.

2.4.1 nb_email_send

```
int nb_email_send (string rcpt, string subj, string msg)
```

The nb_email_send function can be used to send an E-Mail to a particular address. Please note that the E-Mail client must have been set up correctly prior to using this function.

```
rcpt    recipient's email address (e.g. abc@abc.com)
subj    email subject
msg     email content
```

Returns 0 on success or any error code.

2.5 Digital I/O

2.5.1 nb_dio_get

```
int nb_dio_get (string port)
```

The nb_dio_get function retrieves the status of a digital I/O port.

```
port    DIO port to be queried (in1, in2, out1, out2)
```

Returns the DIO status (0 = off, 1 = on) or -1 on error.

2.5.2 nb_dio_set

```
int nb_dio_set (string port, int state)
```

The nb_dio_set function can be used to turn on/off the status of a digital output port.

```
port    digital output port to be configured (out1, out2)
state   new output status (0 = off, 1 = on)
```

Returns -1 on error.

2.5.3 nb_dio_summary

```
string nb_dio_summary (void)
```

The nb_dio_summary function retrieves the status of all digital I/O ports.
Returns a string holding the status of all ports or an empty string on error.

2.6 Configuration

2.6.1 nb_config_get

```
string nb_config_get (string key)
```

The nb_config_get function returns the currently configured value of a particular config parameter.

```
    key         config key (such as sdk.status)
```

Returns the config value or NULL on error.

2.6.2 nb_config_set

```
    int nb_config_set (string config)
```

The nb_config_set function can be used to set system configuration parameters. Note, that this function will immediately apply the values to the system.

```
    config      config to be set in the form key=value (e.g.
       sdk.status=0)
```

Returns -1 on error.

2.6.3 nb_config_summary

```
    string nb_config_summary (void)
```

The nb_config_summary function will return the current system configuration, that is, the difference between the factory and the running configuration.

2.7 Status Information

2.7.1 nb_status

```
    struct nb_status (string section)
```

The nb_status function will return various status values, the following sections can be specified:

```
    section     the status section which shall be queried which can be:
                system          System information
                license         License information
                wwan            WWAN module status
                wlan            WLAN module status
                gnss            GNSS (GPS) module status
```

```
                lan             LAN interface status
                wan             WAN interface status
                openvpn         OpenVPN connection status
                ipsec           IPsec connection status
                dio             Digital IO status
```

Returns a struct holding the relevant status values (see 'status.are' example).

2.7.2 nb_status_summary

```
    string nb_status_summary (void)
```

The nb_status_summary function will return a short summary about the current system status or NULL on error.

2.8 Network Scanning

2.8.1 nb_scan_networks

```
    struct nb_scan_networks (string ifc)
```

The nb_scan_networks function can be used to scan for available networks.

```
    ifc      the interface to scan (e.g. WLAN1 or Mobile1)
```

Returns a struct holding the relevant networks (see examples).

Please note, that scanning a mobile interface will tear down any running WWAN connections. Same applies to WLAN interfaces operating in access-point mode. Therefore the scan interval is limited to 30 seconds.

2.9 File Transfers

The file transfer functions can be used to transfer files from or to a remote server denoted by an FTP or HTTP/HTTPS URL.

2.9.1 nb_transfer_get

```
    int nb_transfer_get (string usr, string pwd, string url, string
       path)
```

The nb_transfer_get function can be used to get a file from a remote server. The usr/pwd arguments can be applied in order to perform authentication.

```
usr the username used for authentication (can be empty)
pwd the password used for authentication (can be empty)
url the URL where to get the file from
path  the local path where the file should be stored
```

Returns -1 on error.

2.9.2 nb_transfer_put

```
int nb_transfer_put (string usr, string pwd, string url, string
   path)
```

The nb_transfer_put function can be used to transfer a file to a remote server. The usr/pwd arguments can be applied in order to perform authentication.

```
usr the username used for authentication (can be empty)
pwd the password used for authentication (can be empty)
url the URL where to put the file
path  the path to the local file which sould be sent
```

Returns -1 on error.

2.9.3 nb_transfer_post

```
int nb_transfer_post (string usr, string pwd, string url, string
   path, string postfields)
```

The nb_transfer_post function can be used to transfer a file to a remote HTTP server. By using the POST method, additional parameters may be passed. The usr/pwd arguments can be applied in order to perform authentication.

```
usr         the username used for authentication (can be empty)
pwd         the password used for authentication (can be empty)
url         the URL where to put the file
path          the path to the local file which sould be sent
postfields      addition POST parameters in the form:
   <key1>=<val1>&<key2>=<val2>&<keyN>=<valN>
```

Returns -1 on error.

2.9.4 nb_transfer_list

```
array nb_transfer_list (string usr, string pwd, string url)
```

The nb_transfer_list function can be used to retrieve the list of files from a remote server. The usr/pwd arguments can be applied in order to perform authentication.

```
usr          the username used for authentication (can be empty)
pwd          the password used for authentication (can be empty)
url          the URL, specifying the directory to be listed
```

Returns an array of struct describing the directory files. They are made up of:
name name of the file size file size in bytes mode file mode and permission (see chmod) user owner username group owner groupname time modification time tm time struct of modification time

2.9.5 nb_transfer_delete

```
int nb_transfer_delete (string usr, string pwd, string url)
```

The nb_transfer_delete function can be used to delete a file from a remote FTP server. The usr/pwd arguments can be applied in order to perform authentication.

```
usr the username used for authentication (can be empty)
pwd the password used for authentication (can be empty)
url the URL containing the path to the to be deleted file
```

Returns -1 on error.

2.10 LED

2.10.1 nb_led_acquire

```
int nb_led_acquire (int led)
```

The nb_led_acquire function will acquire the specified indication LED to be used for a script. Any associated system indication will stop being signalled until the LED is released again. The status LED cannot be acquired/set.

```
led the number of the LED to be acquired
  (starting from left/top) or LED_ALL for all LEDs
```

Returns -1 on error otherwise zero.

2.10.2 nb_led_release

```
int nb_led_release (int led)
```

The nb_led_release function will release an acquired LED again.

```
led the number of the LED to be released or LED_ALL for all LEDs
```

Returns -1 on error otherwise zero.

2.10.3 nb_led_set

```
int nb_led_set (int led, int mode)
```

The nb_led_set function will set the specified LED to a specific mode.

```
led the number of the LED to be released or LED_ALL for all LEDs
mode    the LED mode to be applied which can be specified by
   OR'ing the following colors and types:
  LED_OFF           turn off LED
  LED_COLOR_GREEN   color is green
  LED_COLOR_RED     color is red
  LED_COLOR_YELLOW  color is yellow
  LED_SOLID         type is solid
  LED_BLINK_FAST    type is fast blinking
  LED_BLINK_SLOW    type is slow blinking
```

Returns -1 on error otherwise zero.

2.11 Config/Software Update

The following functions can be used to trigger a configuration or software update of the system. The URL can be specified as follows:

```
http://<server>/<path>      (retrieve file via HTTP)
https://<server>/<path>     (retrieve file via HTTPS)
ftp://<server>/<path>       (retrieve file via FTP)
tftp://<server>/<path>      (retrieve file via TFTP)
file:///<path>              (use local file)
```

Please bear in mind that calling nb_update_software() will result in a system reboot. The nb_update_config() call will restart the SDK which will terminate your scripts. Thus, it is recommended to exit the script after calling this function and check the result later on via nb_update_status().

2.11.1 nb_update_status

```
string nb_update_status (void)
```

The nb_update_status function returns the status of the last or currently running update operation

2.11.2 nb_update_config

```
int nb_update_config (string url)
```

The nb_update_config function will perform a configuration update from the specified URL.

```
url     the URL of the config
```

Returns zero on success.

2.11.3 nb_update_software

```
int nb_update_software (string url)
```

The nb_update_software function will perform a software update from the specified URL.

```
url     the URL of the software
```

Returns zero on success.

2.11.4 nb_update_sshkeys

```
int nb_update_sshkeys (string url)
```

The nb_update_sshkeys function will perform an update of the SSH authorized keys.

```
url     the URL of the keys
```

Returns zero on success.

2.12 Web Pages

The following functions can be used to implement your own pages within the Web Manager. Such a page will appear under the SDK menu as soon as it has been registered.

2.12.1 nb_page_register

```
int nb_page_register (int id, string title)
int nb_page_register (int id, string title, string submenu)
```

The nb_page_register function registeres a new page with the specified identifier and title (as well as submenu).

```
id              identifier
title           page title
submenu         submenu for page
```

Returns -1 on error and otherwise a page structure which can be used for further page functions.

2.12.2 nb_page_unregister

```
int nb_page_unregister (struct page)
```

The nb_page_unregister function can be used to unregister a page again.

```
page    page structure
```

Returns -1 on error, otherwise zero.

2.12.3 nb_page_request

```
struct nb_page_request (struct page)
```

The nb_page_request function listens for incoming requests.

```
page    page structure
```

Returns void on error, otherwise a request structure which holds possible GET and POST parameters.

2.12.4 nb_page_respond

```
int nb_page_respond (struct page, string fmt, ...)
```

The nb_page_responds function can be used to echo back a string to the request and can be called multiple times until page is to be finished. It supports a fmt string and additional arguments that are formatted according to the format string. Please refer to the printf function for more information about format options.

```
page    page structure
```

fmt format string

Returns -1 on error and zero on success.

2.12.5 nb_page_finish

```
int nb_page_finish (struct page)
```

The nb_page_finish function will be used to signal that responding to a request has been finished and data shall be passed to the client.

```
page    page structure
```

Returns -1 on error and zero on success.

2.13 SNMP functions

2.13.1 nb_snmp_traphost

```
int nb_snmp_traphost (string host, int port, int version, string
   community)
int nb_snmp_traphost (string host, int port, int version, string
   user, string password, string auth, string priv)
```

The nb_send_traphost function will set the host for sending SNMP traps.

For an SNMPv1/v2 host the parameters are:

```
host    hostname or address
port    trap port
version   SNMP version (1 or 2)
community community string
```

For an SNMPv3 host the parameters are:

```
host  hostname or address
port  trap port
version SNMP version (3)
user  username
pass  password
auth  authentication protocol (MD5 or SHA)
priv  privacy protocol (DES or AES)
```

Returns -1 on error.

2.13.2 nb_snmp_trap

```
string nb_snmp_trap (string oid, string type, string value)
```

The nb_send_trap function will send an SNMP trap with the specified OID to a remote traphost.

```
oid   SNMP object identifier of the trap
type  type of value to be sent ('e' for empty, 'i' for integer and
   's' for octet string)
value   value to be sent
```

Please note that a traphost has to be set prior to using this function.
Returns -1 on error.

2.14 Various network-related functions

2.14.1 nb_gethostbyname

```
array nb_gethostbyname (string host)
```

The nb_gethostbyname function performs a DNS lookup for the given hostname and returns an array of resolved IP addresses. Please note that a valid DNS is available prior to using this function.

```
host    the to be resolved host
```

Returns an empty array if host could not be resolved.

2.14.2 nb_ifc_address

```
string nb_ifc_address (string interface)
```

The nb_ifc_address function can be used to obtain the first address of an interface.

```
interface       internal interface name (e.g. lan0)
```

Returns the interface address or NULL on error.

2.14.3 nb_ping

```
int nb_ping (string host)
int nb_ping (string host, int timeout)
```

The nb_ping function will send ICMP ping packets to the specified host and returns whether the host correctly responded or not.

```
host            the host to ping
timeout         timeout waiting for a reply (in milliseconds)
```

Returns 1 in case the host is alive, 0 if down and -1 on error.

2.14.4 nb_arp_ping

```
int nb_arp_ping (string host)
```

The nb_arp_ping function will send an ARP request for the specified host and returns whether the host address has been successfully resolved.

```
host            the host address to ping
```

Returns 1 in case the specified host could be resolved, 0 if not and -1 on error.

2.14.5 nb_arp_gratuitous

```
int nb_arp_gratuitous (string interface)
int nb_arp_gratuitous (string interface, string host)
```

The nb_arp_gratuitous function will send an gratuitous ARP advert for the address of the specified interface (or the host address provided). This can be used to update the ARP tables of your neighbors.

```
interface       the interface on which the packet should be send
host            the host address to advert
```

Returns 1 in case the packet could be sent or -1 on error.

2.14.6 nb_etherwake

```
int nb_etherwake (string hwaddr, string interface)
```

The nb_etherwake function will send a WakeOnLan magic packet to wake up sleeping hosts.

```
hwaddr          the Ethernet MAC address of the host
interface       the interface on which the packet is sent
```

Returns 0 in case the packet has been successfully sent or -1 on error.

2.15 Other system functions

2.15.1 nb_syslog

```
int nb_syslog (string fmt, ...)
```

The nb_syslog function creates a message in the system log. Please refer to sprintf for more information about the format string fmt and additional arguments.

```
msg       message to be written to syslog
```

Returns -1 on error.

2.15.2 nb_event_get

```
string nb_event_get (int timeout)
```

The nb_event_get function will poll for system events.

```
timeout         max. number of seconds to wait for an event
```

Returns received event as string or NULL when the specified timeout has been reached.

2.15.3 nb_reboot

```
void nb_reboot (int delay)
```

The nb_reboot function will trigger a system reboot.